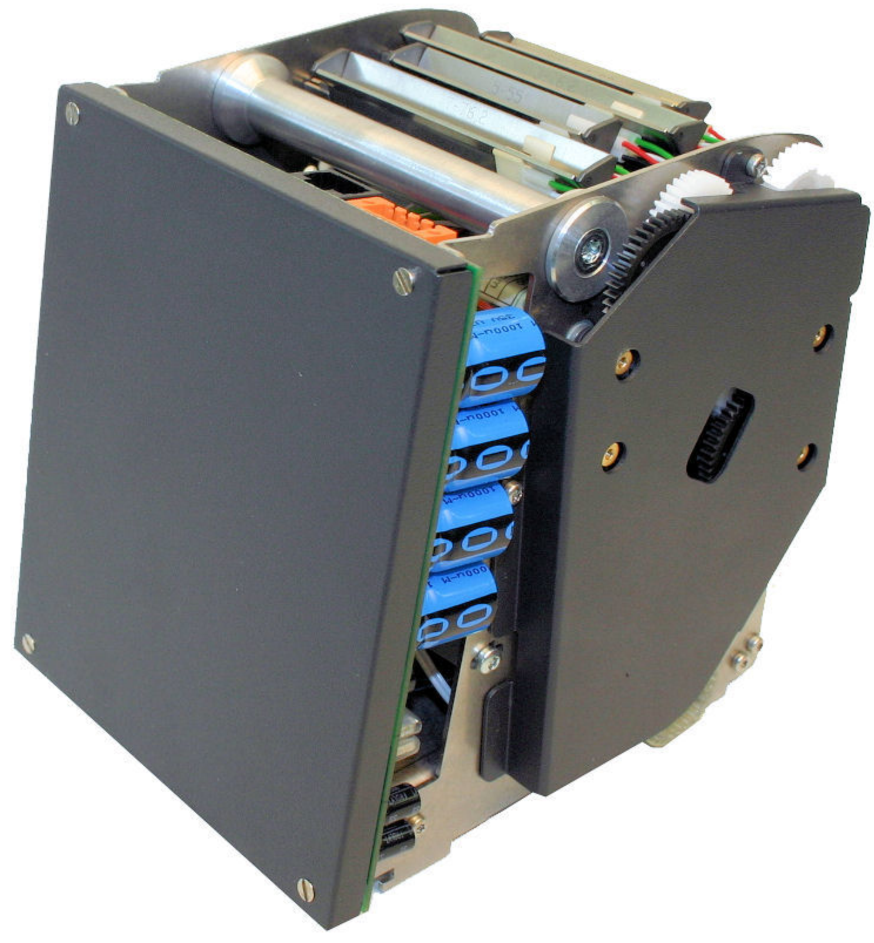


PDM170/V1.04





PRINTON GmbH · Ohrsbergweg 1 · D-69412 Eberbach
Tel: +49 (0)6271 / 96111-0 · Telefax: (0)6271 / 96111-18
e-mail: MAIL@PRINTON-GmbH.de
Internet: www.PRINTON-GmbH.de

Stand: Juli 2007

Inhaltsverzeichnis







Kapitel 1 : Änderungs-Historie

Ver. 1.04

23.07.2007

- Verbesserung des Papiermanagements.
- Festlegen der Tolleranzen für Papiermanagement (Betriebsdaten (Funktion 14)).
- Bugfix bei Ausgaben von Dokumenten die größer als Buffersize sind.

Ver. 1.03

21.06.2007

- Abfrage des verbrauchten Papiers über Steuersequenz <ESC>y9;14;3<CR>, die Angabe erflogt in Millimeter
- Implementieren der Erkennung von „halben" Etiketten Aktivierung über <ESC>k10020;1<CR>.
  Dabei wird anhand der Angaben Etikettenhöhe (Abstand Indexmarken) geprüft ob das ausgegebene Etikett der Angabe entspricht.
- Vollständige Implementierung der Initsequenz <ESC>i.., Parameter w,e wurden hinzugefügt.
- Reduzieren der Basisheizzeit von 450µs auf 400µs
- Erweitern der Testfunktion und Konfiguration für die Sensoren <ESC>y12.. .
  Hier wurde eine Funktion für das Ausmessen der Sensoren implementiert.

Ver. 1.01

04.04.2007

- Implementieren Schnittstelle für Initstring.
- Implementierung Prüfen von Drucken von leeren Tickets über Option k10020;1, nur zusammen mit FPGA vom 05.02.07. Wenn Tickets ausgegeben werden, die nicht Bedruckt sind, wird Fehler 215 gesetzt.
- Bugfix bei Kalibrierung: Bei langen überstehenden Papierstücken konnte es vorkommen das dass Papier nicht geparkt wurde.
- Bugfix Variabel Etiketten: Hier konnte der Variable Anteil des Tickets nicht ausgegeben werden.

Ver. 1.00

09.02.2007

Erstellen der Releaseversion 1.00

- Papier Entwerten mit FF (<ESC>z2<CR> wenn Option k4000 gesetzt ist.
- Überwachung von Papierweg nach Abschneiden. Hier wird die Wegstrecke gemessen wie groß der Abstand zwischen Cutter und Sensor ist, wenn der gemessene Weg mit dem definierten Weg nicht übereinstimmt, dann wird Error #210 abgesetzt.
- Neue Meldungen Nummern im Bereich zwischen #150 und #199 sind keine Warnungen oder Fehler sondern Hinweise. Die Nummern sind:
  - #160 Papier aus Schacht entnommen (Schachtspezifisch)

  - #161 Papier nach Entnahmekommando <ESC>sx-<CR> noch lose im Schacht, kein Schachtwechsel möglich (generelle)
  - #165 Papier leer (früher 204) (Schachtspezifisch)
- Überarbeitung Papierentnahme Problematik .
- Bugfix zu verschiedenen Punkten :
  - Drucker reagiert nicht auf Schachtwechsel
  - Erste Statusabfrage nach <ESC>sx+<CR> meldet noch Status =0000/000, und nicht =0400/000
  - Schachtwechsel wird nicht immer zu Ende ausgeführt.
  - Papier überdrucken
  - Parkieren nach Papierende funktioniert nicht

Kapitel 2 : Allgemeine Beschreibung

Druckerbedienung

Dieses Kapitel soll Ihnen als Hilfe für die Installation des Druckers dienen. Sollten Fragen oder Probleme auftauchen, dann nehmen Sie Kontakt mit Ihrem Lieferanten oder mit der Firma PRINTON GmbH auf. Die Firma PRINTON GmbH bietet Support von 8.30-16.00 Uhr von Montags bis Donnerstag und von 8.30-12.00 Uhr Freitags unter der Nummer 0049-(0)6271-961110.

Allgemeine Produktinformation

Der Einbaudrucker PDM170-X ist ein Thermodirektdrucker zum Bedrucken von blanko Papierrollen, die Rollen können mit einem Klischeedruck bedruckt sein. Bei einer Auflösung von 300 dpi kann eine sehr gute Druckqualität garantiert werden.
Der Drucker ist dafür ausgelegt, dass bis zu vier Papierrollen bedruckt werden können. Hierzu benutzt der Drucker einen integrierten Papierwechsel- mechanismus.

Die notwendige Wartung des Druckers beschränkt sich dank dem anwenderfreundlichen Aufbau auf ein Minimum (Siehe „**Wartung und Repartur Anleitung**“)

Die Ansteuerung des Druckers erfolgt mittels einer Druckersprache per ESC-Sequenzen oder über einen WINDOW-Treiber. Dies erlaubt sowohl eine flexible Gestaltung des Ausdrucks (Text, Barcode, Logos, Linien und Rechtecke), als auch eine umfangreiche Steuerung des Druckablaufs (Druckenergie, Druckgeschwindigkeit usw.).

Der PDM170 Drucker verfügt über eine V24- und USB2.0-Fullspeed Schnittstelle, so wie einen 24V Gleichspannungsanschluß.

Druckerinstallation

Auspacken des Druckers

Jeder Drucker wird auf volle Funktionstüchtigkeit hin überprüft und verpackt. Obwohl die Verpackung extra zur Sicherung des Drucker gegen äußere Schäden konzipiert ist, kann es sein, dass diese bei Transport beschädigt wird. Nach dem Auspacken des Druckers sollte dieser immer sofort auf äußere Schäden überprüft werden. Sollte eine Beschädigung vorliegen, dann ist der folgende Ablauf einzuhalten:

1. Verständigen Sie umgehend den Lieferanten (UPS, Post usw.) von dem Schaden.
2. Stellen Sie dem Lieferanten die gesamte Verpackung zur Verfügung.
3. Unterrichten Sie Ihren Händler von dem Schaden. Weder der Händler noch die Firma PRINTON GmbH haften für solche Schäden.

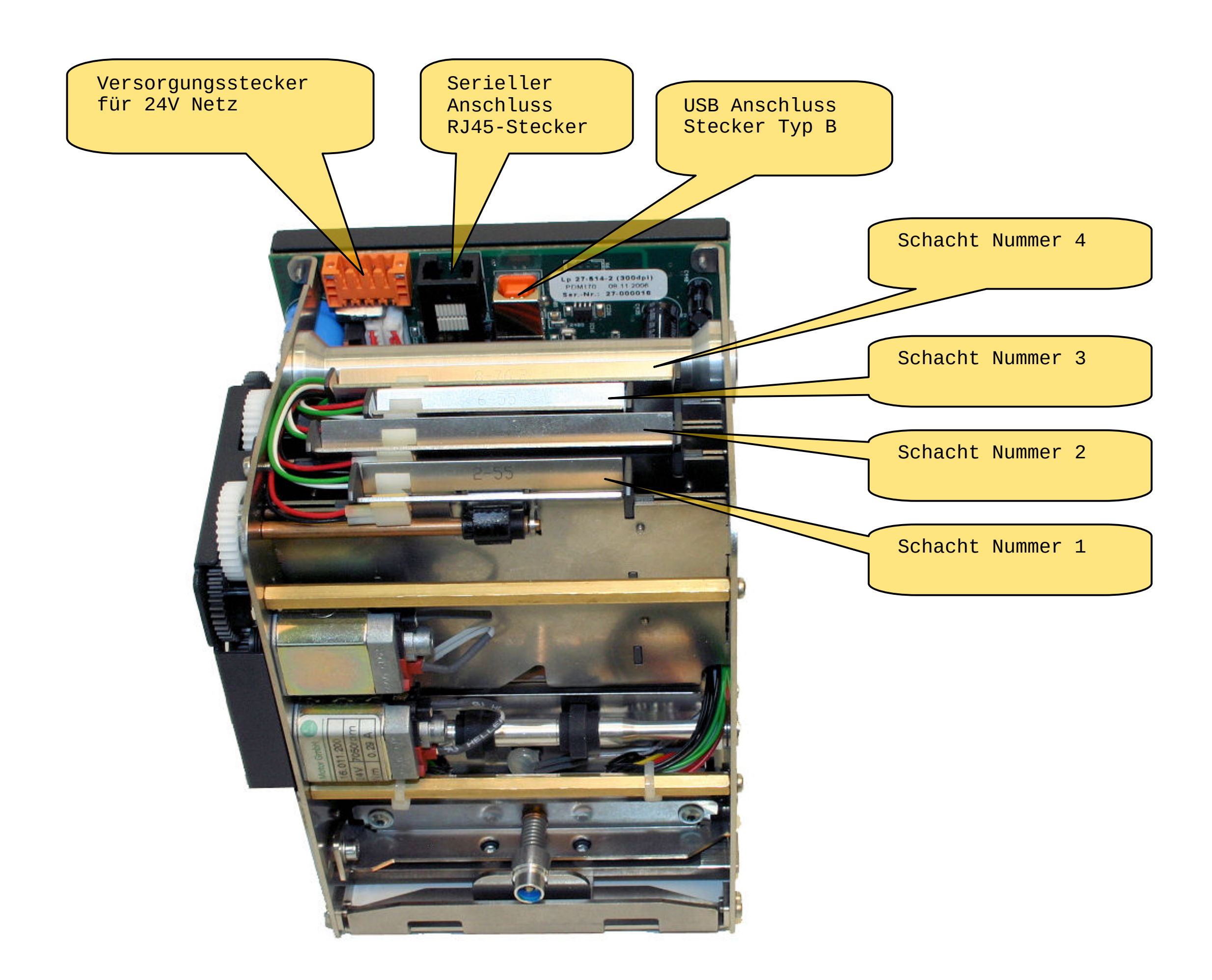
Anschluß des Druckers

Der Drucker wird mit den vorgesehenen Netzkabel an einer Gleichspannung von 24 Volt angeschlossen. Wahlweise kann der Drucker über die V24 Schnittstelle oder mit der USB-Schnittstelle betreiben werden.

Anschluß an den PC

Der Drucker sollte vor dem Anschluß an den PC ausgeschaltet werden. Der Drucker verfügt über eine serielle (V24) und eine USB-Schnittstelle, über die er angesteuert werden kann. Verbinden Sie also den Drucker mit dem erforderlichen Kabel mit dem PC. Nun kann der Drucker eingeschaltet werden.
Verwenden Sie den USB-Anschluss, dann muss ein USB-Treiber installiert werden, siehe hierzu Kapitel „USB-Support“

Druckeraufbau



Versorgungsstecker (Netzbuchse):
An der Stirnseite der Steuerseite befindet sich ein 4-Poliger Netzbuchse. Über diese Buchse wird die Steuerkarte mit einer 24Volt Gleichspannung versorgt.

Schnittstellen:
An der Stirnseite der Steuerkarte befindet sich die Schnittstelle für die V24 Schnittstelle, der Anschluss ist über ein RJ45-Buchse realisiert.
Links neben dem V24 Anschluss befindet sich der Anschluss für den USB-Port.

Dem Anwender stehen standardmäßig eine serielle (V24) und eine USB-Schnittstelle zur Verfügung.
Beide Kabel sind bei der Firma PRINTON GmbH erhältlich.

Bedienelemente :
Direkte Bedienelemente sind nicht vorhanden.

Kapitel 3 : Bedienungs-Anleitung

Einlegen Papier:

Das Druckwerk für den Einbaudrucker ist so ausgelegt, dass neues Papier oder Papierwechsel ohne spezielle Vorkenntnisse oder Spezialwerkzeug durchgeführt werden kann.
Beim Einlegen des neuen Papiers ist darauf zu achten dass die thermosensitive Seite des Papiers in Gegenrichtung der Leiterkarte zeigt.
Für das Einlegen von Papier gibt es eine Steuersequenz: siehe (Schachtsteuerfunktion (<ESC>s..)).

Papier Entnehmen:

Das Druckwerk für den Einbaudrucker ist so ausgelegt, dass neues Papier oder Papierwechsel ohne spezielle Vorkenntnisse oder Spezialwerkzeug durchgeführt werden kann.
Beim Herausnehmen von Papier ist darauf zu achten, das dass Papier komplett aus dem Schacht entfernt wird.
Wurde das Papier komplett entfernt, dann wird der Schacht automatisch geschlossen (Schachtsteuerfunktion (<ESC>s..)).

Einstellung der Papierführung:

Es gibt Druckervarianten mit einer fixen Papierführung und Druckervarianten mit einer variablen Papierführung. Die nachfolgende Beschreibung bezieht sich auf Drucker welche mit einer variablen Papierführung ausgestattet sind.

Das Papier muss immer mittig zur Thermokopfleiste geführt werden. Um dieses immer zu gewährleisten, ist es notwendig die Papierführungslaschen richtig ein zu stellen.
Die Papierführungslaschen können mit einem kleinen Inbusschlüssel geöffnet werden, sind die Laschen geöffnet, dann können diese verschoben werden.
Sind die Laschen auf der richtigen Position, dann müssen die Laschen wieder mit den Schlüssel fixiert werden.
Bei der variablen Papierführung ist es wichtig, das dass Papier gerade an den Drucker zugeführt wird. habe Sie hierzu Fragen, dann wenden Sie sich bitte an die Firma PRINTON GmbH.

Kapitel 5 : Druckerfunktion

Der Drucker kennt eine Vielzahl von Betriebsarten, die per Steuersequenz einstellbar sind.
Nachfolgendes Kapitel beschreibt die Vorgehensweise zu den einzelnen Druckerfunktionen.

Makrofunktionen (Deskriptoren)

Es besteht die Möglichkeit, alle Geräteparameter so abzuspeichern, dass sie nach Ausschalten des Druckers nicht verloren gehen. Der Anwender kann somit ein festgelegten Satz von Geräteeigenschaften erstellen, z.B. Festlegung von Schnittstellenparameter oder aber Einstellungen von Druckparameter für bestimmte Papiersorten, und diese dann abspeichern.

Die Druckeransteuerung verfügt über 4 Makros, welche die Nummer 0 bis 3 tragen. Anhand dieser Makros kann der Drucker auf maximal 4 verschieden festgelegte Konfigurationen zugreifen.

Sobald der Drucker eingeschaltet wird, lädt dieser automatisch das zuletzt verwendete Makro in den Arbeitsbereich. Somit sind dem Drucker bereits beim Einschalten alle Druckerfunktionen bekannt.

Makro 0 wurde als Vorgabe-Makro werksseitig festgelegt. Es ist nicht veränderbar. Durch Laden dieses Makros kann der Drucker immer wieder in den Urzustand zurückgesetzt werden.
Makro 1, 2 und 3 enthalten ebenfalls Vorgabe-Werte wie Makro 0, können jedoch vom Benutzer verändert werden.

Im Auslieferungszustand ist Makro 1 aktiv.

Makro programmieren

Das Verändern von Makros mittels einem Befehlssatz von Steuersequenzen, welche mit folgendem abschließenden Speicherbefehl

<ESC>m*d*<CR> *d* = Makro-Nr. (1-3)

abgeschlossen wird.
Das soeben gespeicherte Makro ist beim nächsten Einschaltvorgang auch das aktive bzw. geladene Makro.

Makro wechseln

Das zuletzt programmierte Makro wird automatisch immer beim Einschalten des Druckers verwendet. Beim Wechseln des Makros mit Hilfe der Steuersequenz
<ESC>m*d*;2<CR> *d* = Makro-Nr. (1-3)
muss der Drucker anschließend neu gestartet werden, bevor das neue Makro aktiv ist.

Bitte beachten Sie, dass der Speichervorgang ca. 3 Sekunden benötigt und in dieser Zeit der Drucker nicht ausgeschaltet werden darf.

Drucker Modi

Der Drucker kann in zwei grundsätzlichen Modi betrieben werden. Die Modi werden ESCAPE-MODUS und MATRIX-Modus bezeichnet.

Die Umstellung der Modi geschieht mit der „Gerätedaten (<ESC>k..)“ Steuersequenz.
Die Kombination der Modi innerhalb eines Druckauftrages ist nicht möglich.
Vor jedem Druckauftrag kann jedoch zwischen den Modi umgeschaltet werden,

ESCAPE-Modus

Generell wird beim ESCAPE-Modus immer ein vollständiges Dokument innerhalb des Druckers aufbereitet, bevor es ausgegeben wird.

Beim Escape-Modus kann mit den „Objektsequenzen“ ein Dokument definiert werden. Der Drucker bereitet intern das Dokument auf.
Dieser Modus hat den Vorteil, dass ein Dokument erst vollständig im Drucker vorhanden ist bevor es ausgegeben wird. Es kann noch vor der Ausgabe entschieden werden ob das Dokument eventuell Fehlerhaft ist. Auch werden relativ wenige Daten über die Schnittstelle Übertragen um ein Ticket zu definieren, Typisch 0,5Kb bis 2Kb Daten.
Der Nachteil ist, das es eine gewisse Zeit dauert bis der Drucker zu drucken beginnt.
Der Modus kann Schachtspezifisch eingestellt werden. Um in den Ecape-Modus zu gelangen muss das Bit7 im Gerätedatenregister (Gerätedaten (<ESC>k..)) zurück gesetzt sein.

Matrix-Modus

Im MATRIX-Mode werden die Druckdaten wie sie empfangen werden sofort ausgegeben.
In diesem Mode ist prinzipiell die maximale Ausgabelänge eines Dokuments nicht begrenzt.
Der Druck beginnt sofort wenn die Daten empfangen werden. Problematisch ist im Matrixmode, wenn der Datenstrom für die Druckdaten aus irgendeinem Grund abreisen sollte. Dann wird das Dokument nur unvollständig gedruckt. Hier werden die Bilddaten komplett an den Drucker übertragen, für ein normales Dokument sind schnell 100Kb Daten erreicht.

Der Modus kann Schachtspezifisch eingestellt werden. Um in den Ecape-Modus zu gelangen muss das Bit7 im Gerätedatenregister (Gerätedaten (<ESC>k..)) gesetzt sein.

Endlospapier

Endlospapier enthält keine Indexmarken zur Synchronisation. Der Drucker muss per Steuersequenz <ESC>k10;0<CR> auf "Endlos" eingestellt sein (= Grundeinstellung).
Papierende wird erkannt. Papierstau kann in diesem Mode nicht festgestellt werden.
Eine Umschaltung von Endlos- auf Etikettenpapier ist auch während des Betriebs mit der Steuersequenz <ESC>k10;1<CR> möglich.

Papier mit Synchronisationsmarken

Bei Papier mit Synchronisationsmarken sind die Papiere untereinander durch eine Synchronisationsmarke gegeneinander getrennt. Der Sensor schaut dabei durch das Synchronisationsloch oder erkennt die „Blackmark“. Der Drucker muss per Steuersequenz <ESC>k10;1<CR> auf "Etikett" eingestellt sein.
Wird eine Blackmark für Synchronisation benutz, dann muss das Flag <ESC>k20;1<CR> zusätzlich gesetzt sein.

Es wird grundsätzlich auf das Etikettenende synchronisiert. Vor dem nächsten Druckvorgang erfolgt ein Vorschub zum Anfang des nachfolgenden Papierabschnitts.

Papierweg Überwachung:

Der Drucker unterstütze eine Überwachung des Papierwegs.
Die Überwachung von Papierweg prüft die korrekte Länge des Papiers nach dem Abschneiden.
Hier wird die Wegstrecke gemessen wie groß der Abstand zwischen Cutter und Sensor ist, wenn der gemessene Weg mit dem definierten Weg nicht übereinstimmt, dann wird Error #210 abgesetzt. Hier wird eine Toleranz von ca. +/- 2,5 mm toleriert.

Statusmeldung

Eine Statusanforderung ist über V24 möglich.
Der Statusreport für Kurzstatus kann jederzeit angefordert werden. Wird gerade ein Druckauftrag ausgeführt, sendet der Drucker eine reduzierte Statusmeldung. Jede Ausgabezeile der Statusmeldung wird mit <CR><LF> beendet.

Lang-Statusanforderung, falls <ESC>!<ENQ> (<ENQ> = 05h bzw. CTL/E):

PDM170-Vx.xx/eee/S	→ Programm-Version x.xx – Programm Versionsnummer „0..9“ /eee - Extention „a..z“ (kann auch entfallen) /S - Subtype „A..Z“ (kann auch entfallen)
=0000	→ Druckerstatus (Hexadezimal)
#0	→ Noch zu druckende Etikettenanzahl des akt. Druckauftrages
***65536**	→ Noch verfügbarer Eingabespeicher
>12345678	→ Barcode der zuletzt gelesen wurde. Diese Zeile wird nur ausgegeben wenn das entsprechende Flag in den Gerätedaten (<ESC>k..) gesetzt ist.
/000	→ Fehlercode (höchste Priorität) anzeigen und löschen.

Der Druckerstatus besteht aus einem 16 Bitlangen Wert. Dabei kommen den einzelnen Bits im gesetzten Zustand (=1) bestimmte Bedeutungen zu:

Statusword

Bit0	(0001h)	Zeigt an, dass der Drucker bereit ist Steuersequenzen zu akzeptieren. Bei Programmstart oder nach jedem Druck wird dieses Bit zurück gesetzt. Mit der Steuersequenz <ESC>!<BS> kann die Kommunikation freigegeben werden.
Bit1	(0002h)	Zeigt einen Drucker-Reset an. Wird nur mit der ersten Kurz-Statusabfrage angezeigt und gelöscht.
Bit2	(0004h)	Zeit an, dass ein Etikett gedruckt wurde. Das Bit wird vor dem nächsten Druckvorgang automatisch zurückgesetzt.
Bit3	(0008h)	Der aktuelle Druckauftrag ist noch nicht vollständig ausgeführt. Dies ist dann der Fall, solange die zu druckende Etikettenanzahl noch > 0 ist.
Bit4	(0010h)	Ein gedruckte Etikett wurde geschnitten und kann entnommen werden. Das Bit wird vor dem nächsten Druckvorgang automatisch zurückgesetzt.

Bit5	(0020h)	Es liegt ein Datensatz aufbereitet im Drucker vor. Somit kann ein Druckkommando jederzeit ausgeführt werden. Das Bit wird nur während einer Druckaufbereitung gelöscht oder aber bei Auftragsstornierung.
Bit6	(0040h)	Zeigt an, dass der Transfer für ein Etikett im Augenblick aktiv läuft. Dieses Bit wird mit dem Empfang von Start of Text (STX) gesetzt. Mit End of Text (ETX) wird es zurück gesetzt.
Bit7	(0080h)	Zeigt an, dass der Papiertransport noch aktiv ist.
Bit 8..9	(0300h)	Zeigen die aktuelle Schachtnummer an.

Bit9	Bit8	
0	0	Schacht1
0	1	Schacht2
1	0	Schacht3
1	1	Schacht4

Bit10	(0400h)	Zeigt an, dass ein Schachtwechsel aktiv ist.

Statusbits können natürlich auch kombiniert auftreten, z.B. 00111000 (0038h).

Kurzstatus, falls <ESC>!<ACK> (<ACK> = 06h bzw. CTRL/F):

=0000/000 → Druckerstatus /
Fehlercode (höchste Priorität) anzeigen und löschen

Die Statusanforderung kann auch während des Drucks durchgeführt werden:

Während des Drucks darf nur alle 100ms eine Statusanforderung an den Drucker gesendet werden.

Software-Reset

Die Steuersequenz! <ESC>!! bewirkt einen Drucker-Reset. Dabei wird der Drucker wieder neu gestartet. Die noch im Eingabespeicher befindlichen Daten werden dabei gelöscht.

Die Steuersequenz <ESC>!<CTL/G> bewirkt ebenfalls ein Rücksetzen des Druckers, jedoch erfolgt dabei kein Papiervorschub und kein Schneidevorgang. Speicherinhalte gehen verloren.

Error-Recover

Beim Auftreten eines Fehlers bei Schachtwechsel oder bei Papierstau, gibt es noch die Möglichkeit ein Error-Recover durchzuführen.

Ein Error-Recover bewirkt, das der Schacht bei dem der Fehler aufgetreten ist nochmaligst ein Papierkalibrierung durchgeführt wird. Ist die Papierkalibrierung ohne Fehler abgelaufen, dann werden alle weiteren Schächte die mit Papier bestückt sind getestet.

Ein Error-Recover wird mit der Steuersequenz: <ESC>!<HT> gestartet.

Kapitel 6 : Datensatzaufbau

Symbole und Konventionen

Im weiteren Verlauf der Beschreibung werden einige besondere Symbole und Konventionen angewandt:

Fettgedruckte Zeichen sind Schlüsselwörter und müssen genauso eingegeben werden.
Kursivschrift innerhalb einer Sequenz stellen Platzhalter dar, die der Anwender durch bestimmte Angaben ersetzten muss.
<u>Unterstrichene</u> Werte sind Standard-Werte, welche sich auf einen 300 DPI-Drucker beziehen.
Optionale Eingabemöglichkeiten werden in Rechteckklammern [ ] dargestellt.
Nichtdruckbare Steuerzeichen werden in spitzen Klammern eingeschlossen, z.B. <CR>.

Die Eingabe von Steuerzeichen wird in spitzen Klammer dargestellt und ist zwingend.
An dessen Stelle muss der entsprechende ASCII-Wert gesendet werden:

```
<STX>  ⇨  02 Dezimal, 02 Hexadezimal, CTRL/B
<EOT>  ⇨  04 Dezimal, 04 Hexadezimal, CTRL/D
<ENQ>  ⇨  05 Dezimal, 05 Hexadezimal, CTRL/E
<ACK>  ⇨  06 Dezimal, 06 Hexadezimal, CTRL/F
<BEL>  ⇨  07 Dezimal, 07 Hexadezimal, CTRL/G
<BS>   ⇨  08 Dezimal, 08 Hexadezimal, CTRL/H
<CR>   ⇨  13 Dezimal, 0D Hexadezimal, CTRL/M
<ESC>  ⇨  27 Dezimal, 1B Hexadezimal, CTRL/[
```

Für Zahlenwerte im ASCII-Eingabeformat gelten folgende Konventionen:
d bedeutet: Dezimalzahl
h bedeutet: Hexadezimalzahl
Ist die Eingabe an ein festes Format gebunden, so wird dies mit Großbuchstaben angezeigt:
HHHH steht z.B. für 4-stellige Hexadezimalzahl

Für Charakterwerte gelten folgende Konventionen:
c bedeutet: 1 Charakter (A - Z, a - z)
a bedeutet: 1 Charakter (A - Z, a - z, 0 - 9)

0 | 1 stellt 0 oder 1 als Eingabemöglichkeit zur Verfügung.

Tief$_{\text{gestellte}}$ Texte dienen lediglich zur Information:
D_{typ} bedeutet: 1-stellige Typnummer, dezimal.

Benutzung von Steuersequenzen

Steuersequenzen nehmen Einfluss auf die Gerätekonfiguration und den Steuerungsablauf des Druckers.

Eine Steuersequenz hat folgenden Aufbau:

- Beginn mit <ESC> und einen darauffolgenden Kleinbuchstaben als Schlüsselwort sowie einem oder mehrere Parameter.
- Eine Steuersequenz endet mit einem darauffolgendem Controlzeichen, z.B. <CR> oder <ESC>.
- Die Summe aller aufeinanderfolgenden Steuersequenzen bilden den Steuerblock.

Alle Steuerparameter haben bereits eine Grundeinstellung. Steuersequenzen ohne Angabe eines Parameters liefern im Allgemeinen ihren Wert zurück.

Benutzung von Objektsequenzen

Objektblöcke bestimmen das eigentliche Druckbild, nachdem die Gerätekonfiguration abgeschlossen ist.

Objektblöcke haben folgenden Aufbau:

- Ein Objektblock besteht aus Objektsequenzen, die mit <ESC> und einem darauffolgendem Großbuchstaben als Schlüsselwort beginnen.

- Eine Objektsequenz endet mit einem darauffolgendem Controlzeichen, z.B. <CR> oder <ESC>.
  Maximal können 64 variable Objektsequenzen pro Layoutblock verwendet werden.
  Bei variable Barcodeobjekte mit Subscriptzeile werden zwei Objektsequenzen benötigt.
  Die Anzahl der fixen Objektsequenzen ist unbegrenzt.

- Alle Objektblöcke zusammen bilden einen Layoutblock. Ein Layoutblock muss zur Unterscheidung Steuerblöcken in <ESC><STX>....<ESC><EOT> eingeschlossen werden.

- Variable Objektsequenzen werden formal wie Steuersequenzen behandelt.

Alle Objektparameter haben bereits eine Grundeinstellung, z.B. Zeichenabstand = 1 Dot.

Datensatzaufbau für ESCAPE-MODUS

Der Datensatz hat folgenden Aufbau:

```
Datensatz       = { Steuerblock, Layoutblock, Steuerblock }

Steuerblock     = { Steuersequenz 1 [Steuersequenz n] }          n=[2...]
Steuersequenz   = { <ESC >x.... <CR>}                            x=[a|b|...|z]

Layoutblock     = { <ESC ><STX> Objektblock 1 [Objektblock n]
                    <ESC ><EOT> }                                n = [2...]
Objektblock     = { [Objektsequenz n]  Objekt }                  n = [1...]
Objektsequenz   = { <ESC> X ... [<CR>] }                         X = [A|C|...|Z]
Objekt          = { <ESC> B|T|L|X ... [<CR>] }
```

Datensatz
Beschreibt das komplette Drucklayout einschließlich Druckerkonfiguration und Druck-vorgang.

Steuerblock
Beschreibt die komplette Druckerkonfiguration und Druckvorgang.

Steuersequenz
Setzt einen Steuerparameter für Druckerkonfiguration bzw. Druckvorgang.

Objektblock
Beschreibt das komplette Kartenlayout.

Objektsequenz, Objekt
Das Druckerlayout beinhaltet Datenfelder unterschiedlichster Art. Eine Objektsequenz liefert einen Parameter zur Spezifikation dieses Feldes. Ein *Objekt* ist formal eine Objektsequenz und spezifiziert den Feldtyp (Text, Barcode, Logo/Linie/Rahmen). Diese muss immer an letzter Stelle in einem Objektblock stehen. Die vorhergehenden Objektsequenzen beziehen sich auf dieses Objekt.

Etikettensätze werden zunächst im Eingabespeicher zwischengespeichert und nacheinander aufbereitet.
Ein bereits aufbereiteter Datensatz belegt keinen Eingabespeicher mehr.

Datensatzaufbau für MATRIX-MODUS

Der Datensatzaufbau für den Matrixmodus unterscheidet sich zwischen den Datensatzaufbau des ESCAPE-Modus nur im Bereich des Layoutblocks.
Im Layoutbock sind nur zwei Objektsequenzen (Bildhintergrund) erlaubt.
Alle anderen Objektsequenzen führen zu einem Fehler.



Beispiel für einen Datensatz

Beginn des Datensatzes

Steuersätze: (1. Steuerblock)

```
<ESC>k1;0<CR>
```

Geräteparameter: Drucken ohne Print-Taste

```
<STX>
```

Beginn Layoutblock

Objektblock 1 (=Barcodesatz)

```
<ESC>I35<CR>
<ESC>G150<CR>
<ESC>R0<CR>
<ESC>BEAN13;H60;B3;P1>401234567890<CR>
```

Objektblock 2 (=Textdaten)

```
<ESC>I35<CR>
<ESC>G20<CR>
<ESC>R270<CR>
<ESC>D1<CR>
<ESC>C2<CR>
<ESC>F3<CR>
<ESC>TARIAL18F;Drehung 270°<CR>
```

Text = ' Drehung 270° '

```
<EOT>
```

Ende Layoutblock

Steuersätze: (2. Steuerblock)

```
<ESC>#1<CR>
```

Druckauftrag

Ende des Datensatzes

Diagramm Datenfluss für einen Druckdatensatz in ESC-Mode

Datenfluß Übertragung Telegramm für ESC-Mode

Begin

Abfrage Betriebsdaten:
<ESC>y9;14;0<CR>
Anzahl Schnitte,
Betroh
mungen,
..

Fehlernummer < 200

Status Abfrage.
<ESC>!<ENQ><CR>

Fehlernummer <> 0

Fehler Typ:

Fehlernummer >= 200

Fehler Handling,
Papier leer,
Hardwarefeher, ..

Steuer Block

Konfiguration an Drucker Senden
(Escape die mit <ESC>[Kleinbuchstabe] beginnen.
Escape-Mode: <ESC>k800;0<CR>
Synchronisation : <ESC>k30;1<CR>
Höhe : <ESC>b600<CR>
usw.

Layoutblock

Start Layoutblock:
<ESC>[STX]<CR>
Innerhalb vom Layoutblock sind nur
Steuersequenzen erlaubt die mit
<ESC>[Großbuchstabe] .. beginnen

Senden Layoutdaten:
<ESC>I35<CR><ESC>G150<CR><ESC>R180<CR><ESC>TARIAL08F;TextText<CR>
<ESC>I35<CR><ESC>G200<CR><ESC>R0<CR><ESC>D2<CR><ESC>TARIAL08F;Text2Text2<CR>
<ESC>I35<CR><ESC>G250<CR><ESC>R0<CR><ESC>TARIAL10F;Text3Text3<CR>
usw.

Prüfen au f Fehler
Generierung von Fehler
Nr. 31 .. 69

Ende Layoutblock

Ende Layoutblock:
<ESC>[EOT]<CR>

Steuer Block

Status Abfrage.
<ESC>!<ENQ><CR>

Status:
Bit2 ist gesetzt
-> Druckbestätigung
Bit 4 ist Null
-> Nicht geschnitten
Bit 5 ist gesetzt (20h),
Fehler sollte keiner vorliegen,
sonst Daten überprüfen.

Druckkommando erteilen
<ESC>#1>CR>

Status Abfrage.
<ESC>!<ENQ><CR>

Status:
Bit2 ist gesetzt
-> Druckbestätigung
Bit 4 ist gesetzt
-> Schnittbestätigung
Bit 5 ist gesetzt (20h),
Fehler sollte keiner vorliegen,
sonst Daten überprüfen.

Abfrage Betriebsdaten:
<ESC>y9;14;0<CR>
Anzahl Schnitte,
Betrohmungen,
..

Fehlernummer
Pruefen

Fehlernummer <> 0

Fehler Handling,
Papier leer,
Hardwarefeher, ..

Druck Ende

Diagramm Datenfluss für einen Druckdatensatz in Matrix-Mode

Druck im Matrixmode

Begin

Status Abfrage.
<ESC>!<ENQ><CR>

Fehler Typ:

Fehlernummer >= 200

Fehlernummer <> 0

Fehler Handling,
Papier leer,
Hardwarefeher, ..

Steuer Block

Konfiguration an Drucker Senden
(Escape die mit <ESC>[Kleinbuchstabe] beginnen.
Synchronisation : <ESC>k30;1<CR>
Abschneiden nach Druck : <ESC>k100;1<CR>
Höhe : <ESC>b600<CR>
usw.

Statusbyte: 14/00
Bit 2 -> X
Bit 3 -> 0
Bit 4 -> X
Bit 5 -> 0
Bit 6 -> 0
Bit 7 -> 0

Layoutblock

Start Layoutblock:
<ESC><STX><CR>
Innerhalb vom Layoutblock sind nur Steuersequenzen erlaubt die mit <ESC>[Großbuchstabe] .. beginnen

Statusbyte: 60
Bit 2 -> 0
Bit 3 -> 0
Bit 4 -> 0
Bit 5 -> 0
Bit 6 -> 1
Bit 7 -> 0

Senden Layoutdaten:
Bildhintergrund (Reasterdaten)
<ESC>Y...<CR>
<ESC>Y...<CR>
<ESC>Z10<CR>
<ESC>Y...<CR>
<ESC>Y...<CR>
usw.

Statusbyte: E8
Bit 2 -> 0
Bit 3 -> 1
Bit 4 -> 0
Bit 5 -> 1
Bit 6 -> 1
Bit 7 -> 1

Ende Layoutblock

Ende Layoutblock:
<ESC><EOT><CR>

Statusbyte: A8
Bit 2 -> 0
Bit 3 -> 1
Bit 4 -> 0
Bit 5 -> 1
Bit 6 -> 0
Bit 7 -> 1

Steuer Block

Vorschub und Schneiden

Statusbyte: 8C
Bit 2 -> 1
Bit 3 -> 1
Bit 4 -> 0
Bit 5 -> 0
Bit 6 -> 0
Bit 7 -> 1

Papier auf Idel

Statusbyte: 94
Bit 2 -> 1
Bit 3 -> 0
Bit 4 -> 1
Bit 5 -> 0
Bit 6 -> 0
Bit 7 -> 1

Status Abfrage.
<ESC>!<ENQ><CR>

Fehlernummer
Pruefen

Fehlernummer <> 0

Fehler Handling,
Papier leer,
Hardwarefeher, ..

Druck Ende

Statusbyte: 14
Bit 2 -> 1
Bit 3 -> 0
Bit 4 -> 1
Bit 5 -> 0
Bit 6 -> 0
Bit 7 -> 0

Kriterien für Druckbestätigung:

Um zu entscheiden ob ein Druck ausgeführt wurde, gibt es verschiedene Kriterien.
Das Wichtigste Kriterium ist das Statusflag in der Statusmeldung
Das Bit 2 zeigt an ob ein Druck soweit zuende geführt wurde, dass ein Dokument komplett bedruckt wurde.
Das Bit 4 zeigt an ob ein Dokument komplett geschnitten wurde und somit ausgegeben wurde.

Sind die Flags nicht gesetzt, dann ist über die Fehlernummer zu Prüfen, welches Problem zu der fehlerhaften Ausgabe des Dokuments geführt hat (Time Out Daten, Papierfehler, Papierende, usw.)



Kapitel 7 : Steuersequenzen

Aufbau:

Steuerblock	= { Steuersequenz 1 [Steuersequenz n] }	n=[2...]
Steuersequenz	= { <ESC>x.... <CR>}	x=[a|b|...|z]

Auflistung der möglichen Steuersequenzen:

Druckkommando

Bildhöhe

Bildbreite

Spendeposition

Schneideposition

TOF-Offset

Gerätedaten

Druckgeschwindigkeit

Ländercode

X-Bildversatz

Heizzeit Thermokopf

Makrosteuerung

Konfiguration Etikettentimeout

Konfiguration Etikettenzähler für Vorende

Direkte Steuerbefehle

V24-Parameter

Sondersequenzen

Auf den nachfolgenden Seiten sind die einzelnen Steuersequenzen detailliert beschrieben.

Read Back Funktion

Alle Steuersequenzen liefern ihren aktuellen Wert zurück, wenn sie ohne Parameter an den Drucker gesendet werden.
Beispiel: <ESC>j<CR> liefert einen Zahlenwert zwischen 20 und 200.

Druckkommando

Steuersatz:	<ESC>#*d*<CR>
Beispiel:	<ESC>#10<CR>

Parameter:
d = Etikettenanzahl (0...9999)

Das Druckkommando startet einen Druckauftrag mit einer angegeben Anzahl Etiketten unter der Voraussetzung, dass ein Datensatz aufbereitet im Drucker vorliegt.

Die Sequenz <ESC>#0<CR> storniert einen Druckauftrag einschließlich des vorliegenden Datensatzes, z.B. wegen eines aufgetretenen Fehlers.

Bildhöhe

Steuersatz:	<ESC>**b***d*<CR>
Beispiel:	<ESC>b240<CR>

Parameter:
d = Bildhöhe (150...1680...3996) * 1/11,8mm (Anzahl Motorschritte)

Die Steuersequenz legt die Höhe des Bildbereiches (Image) fest. Sie ist in der Regel identisch mit der Etikettenhöhe.

Bei **Endlospapier** legt die Bildhöhe den exakten Transportweg fest. Papierstau kann nicht festgestellt werden.

Bei Etiketten bestimmt nach wie vor die Etikettenlänge den Transportweg, falls die Bildhöhe nicht länger als das Etikett ist.
Ist die Bildlänge > Etikettenlänge, wird über mehrere Etiketten gedruckt und sich anschließend auf das nachfolgende unbedruckte Etikett synchronisiert.

Die Bildlänge < Etikettenlänge macht keinen Sinn, solange es keine Speicherprobleme gibt. Außerdem verringert sich die Transportgeschwindigkeit außerhalb des Bildbereiches auf 75mm/s.
Papierstau wird nach etwa 241mm festgestellt, wenn keine Indexmarke erkannt wurde.

Bildbreite

Steuersatz:	<ESC>**c***d*<CR>
Beispiel:	<ESC>c800<CR>

Parameter:
d = Bildbreite (64...960...1280) * 1/11,8mm

Die Steuersequenz legt die Breite des Bildes (Image) vor. Sie wird jedoch immer vom Drucker um ein Vielfaches von 16 aufgerundet, so dass sich die Bildbreite im 1.3mm-Raster verändern lässt.

Der Wert für die Bildbreite kann jederzeit kleiner als das tatsächliche Etikettenformat sein. Der resultierende Druckbereich wird immer linksbündig verschoben.

Die maximale Bildbreite, die Eingegeben werden kann, ist abhängig von der im Drucker eingebautem Thermoarray. Ein Thermoarray mit einer Druckbreite von 81,3mm bei 300 DPI hat zum Beispiel maximal 960 Dots. Thermoarrays mit 100mm haben eine Breite von 1280 Dots.

Spendeposition (nur in Verwendung mit Labeltaken)

Steuersatz:	<ESC>**d***d*<CR>
Beispiel:	<ESC>d48<CR>

Parameter:
d = Spendeposition (0...100...600) * 1/11,8mm

Das Etikett wird zur Entnahme oder aber auch nur zur Sichtkontrolle nach Druckende weiter heraustransportiert.

Schneideposition

Steuersatz:	<ESC>e*d*<CR>
Beispiel:	<ESC>e20<CR>

Parameter:
d = Schneideposition (+/- 190) * 1/24mm

Mit der Schneideposition kann nochmaligst eine Feinjustage der Schnittposition durchgeführt werden.
Es ist zu Beachten, dass der Parameter nicht zu groß gewählt wird, da sonst das Papier unter umständen so weit zurück transportiert wird, dass es nicht mehr durch die Transportrollen erfasst wird.

Druckgeschwindigkeit

Steuersatz:	<ESC>**j***d[;d_1]*<CR>
Beispiel:	<ESC>j125<CR>

Parameter:
d = Druckgeschwindigkeit in mm/s (50...125...150mm/s)

Parameter d_1 nur möglich bei Mehrformatdrucker.
d_1 = Wechselgeschwindigkeit der Papierformate (20...150...200mm/s).

Die *Druckgeschwindigkeit* legt die Transportgeschwindigkeit des Etikettenpapiers innerhalb des Bildbereiches fest. Sie beeinflusst die Druckqualität.

Gerätedaten (<ESC>k..)

Steuersatz:	<ESC>**k***d*[**;0**|**;1**]<CR>
Beispiel:	<ESC>k1;1<CR>

1. **Geräteparameter:**

d = 1 - Drucker-Taste (FLAG_PRINT_TASTE)
d = 2 - Label Taken (FLAG_LABEL_TAKEN)
d = 10 - Synchronisation (FLAG_INDEX)
d = 20 - Reflexlichtschranke für Synchronisation (sonst Durchlicht)(FLAG_REFLEX)
1 = Reflexlichtschranke
0 = Durchlicht

d = 80 - Abschneider vorhanden (FLAG_ABSCHNEIDER)
d = 100 - Abschneiden (FLAG_ABSCHNEIDEN),
Werden Druckdaten im Matrixmode an den Drucker gesendet, wird nach dem die Daten übertragen wurden und Gedruckt wurden, das Etikett sofort geschnitten.
d = 200- Abschneiden nach Auftrag (FLAG_ABSCHNEIDEN_NACH_AUFTRAG)
d = 800- Druckmodus (FLAG_MATRIXMODE) (Nur PDM170)
1 = Drucker befindet sich im Matrixmode
0 = Drucker befindet sich im Escapemode
d = 2000 - Barcodeleser (FLAG_BARCODELESER)
Ist dieses Flag gesetzt, dann wird der Interne Barcodeleser verwendet. Gleichzeitig wird die ***Statusmeldung*** *um die Barcodezeile erweitert.*
d = 4000- Entwerten Ist diese Flag gesetzt, dann wird beim Form Feed <ESC>z2<CR> das Papier überdruckt.

Die Steuersequenz ermöglicht die Einstellung verschiedener Geräteparameter.
Ohne Schalter werden alle Parameter gleichzeitig gesetzt bzw. rückgesetzt.
Dabei werden die Wertigkeiten der einzelnen Positionen addiert.

Beispiel:

<ESC>k1<CR> Alle Parameter rücksetzen, jedoch
- Druckvorgang nur über die Drucker-Taste

Mit dem optionalen Schalter werden gezielt einzelne oder mehrere Parameter gesetzt bzw. einzelne oder mehrere Parameter rückgesetzt.



d = 10;0 Endlospapier (keine Synchronisation)
d = 10;1 Etiketten synchronisieren auf Durchlicht

d = 30;0 Endlospapier (keine Synchronisation)
d = 30;1 Etiketten synchronisieren auf Blackmark

Durch Addition der einzelnen Wertigkeiten können mehrere Einstellungen gleichzeitig vorgenommen werden.

Beispiele:

<ESC>k30;0<CR> Alle Parameter bleiben unverändert, jedoch

- Etikettensynchronisation ausschalten
- Reflex ausschalten

2. Erweiterte Geräteparameter:

Über d>10000 können erweiterte Gerätedaten eingestellt werden. Die erweiterten Gerätedaten werden getrennt von den oben beschrieben Gerätedaten behandelt.

d = 10010 Test von Überstehenden Papier bei Systemstart. Wird Papier gefunden, dann wird es Entwertet und abgeschnitten.
1 => Test aktiv.

d = 10020 Plausibilitätsprüfungen aktivieren.
Hierbei werden mehrere Prüfungen aktiviert.

1. Prüfung: Auf Ausgabe von Blankopapier.
   Hierbei wird geprüft ob bei der Ausgabe eines Fahrausweises, Belegs Druckdaten an den Thermokopf übermittelt wurden. Sind keine oder nur sehr wenige Daten übermittelt worden, dann wird Error 215 ausgegeben.
2. Prüfung: Cut und auf Sync. fahren
   Hierbei wird der Abstand gemessen, wenn der Abstand nicht stimmt, dann wird Error 211 ausgegeben.
3. Prüfung von Etiketten mit Synchonisationsmarken.
   Ist in der Initsequenz der Parameter h-für die Etikettenhöhe gesetzt, dann wird beim Spenden eines Etiketts die Spendelänge mit der gesetzten Höhe verglichen, stimmt die Höhe nicht, dann wird der Fehler 210 ausgegeben.

1 => Test aktiv.

d = 10100 Selbsttest für Abschneider beim Programmstart unterdrücken
1 => Abschneider nicht Testen.

d = 10200 Korrektur Schneidepostion. Hierbei wird die Schneideposition entsprechend dem Wert von <ESC>t[PARAMETER]<CR> geändert.
Ist dieses Flag gesetzt, dann ist der Wertebereich für <ESC>t entsprechende erweitert.

d = 18000 *Systemtimeout* aktivieren (30sec)

Durch Addition der einzelnen Wertigkeiten können mehrere Einstellungen gleichzeitig vorgenommen werden.

Beispiel:

<ESC>k10010<CR>	Alle erweiterten Geräteparameter werden zurückgesetzt, jedoch ■ Papierprüfung ist aktiviert

Readback Ausgabe, wenn Papierprüfung aktiviert ist:

```
0180  10010
```

Ländercode

Steuersatz:	<ESC>**n***d_1*[**;***d_2*]<CR>
Beispiel:	<ESC>n1;2<CR>

Parameter d_1:

0 ANSI
1 ASCII
2 Multilingual

weitere Parameter d_2, falls ASCII:

d = 0 USA
d = 1 England
d = 2 Deutschland
d = 3 Dänemark
d = 4 Frankreich
d = 5 Schweden
d = 6 Italien
d = 7 Spanien
d = 8 Norwegen
d = 9 Holland

Die Steuersequenz ermöglicht die Änderung der voreingestellten Zeichentabelle. Beim ASCII-Zeichensatz können zusätzlich Ländercodes festgelegt werden.

TOF-Offset

Steuersatz:	<ESC>**t***d*<CR>
Beispiel:	<ESC>t20<CR>

Parameter:

d = TOF-Offset (+/- 50) * 1/24mm t=0
Mit Parameter k10200
d = TOF-Offset (+ 200, -50) * 1/24mm t=0

Mit diesem Parameter kann die Offsetposition gegenüber der TOF-Position erreicht werden.
TOF-Position ist die Position relativ zur Papierkante, bei Null wird genau auf Papierkante begonnen zu Drucken.
Dieses ist dann Sinnvoll, wenn die Schnittposition bei Indexsteuerung verschoben werden soll.

Ist die Option k10200 im erweiterten Optionsregister gesetzt, dann kann mit der TOF - Position in einem größeren Bereich der Wert geändert werden.
Ist Parameter gleich Null dann ändert sich nichts, ist der Parameter ungleich Null, dann wird das Papier vor Druckbeginn um den Eingestellten Wert weniger zurück gezogen. Dieses bedeutet, dass am Anfang des Drucks entsprechend vom Parameter ‚d' eine unbedruckte Lücke bleibt.

X-Bildversatz

Steuersatz:	<ESC>**u***d*<CR>
Beispiel:	<ESC>d48<CR>

Parameter:
d = X-Bildversatz (-50...0...49) * 1/12mm

Das Bild wird soweit wie möglich nach rechts (pos. Werte) bzw. nach links (neg. Werte verschoben. Es kann nicht über den Rand hinausgeschoben werden.

Thermokopf-Heizzeit

Steuersatz:	<ESC>**w**[**+**|**-**]*d;[d_1]*<CR>
Beispiel:	<ESC>w-5<CR>

Parameter:
d = Heizzeit-Veränderung (-50 ... 0 ... +30)
d1 = Offset für Änderung der Historyheizzeiten, sofern History aktiv ist (-30 ... 0 ... +10).

Die Steuersequenz ermöglicht die Veränderung des Schwärzungsgrad durch Erhöhen bzw. Erniedrigen der Thermokopf-Heizzeit zwischen -50% und +30%.
Der Histroyheizeit Offset bezieht sich auf alle Schächte gleichwertig, wobei der Heizzeitwert (d) für jeden Schacht individuell ist.

Makrosteuerung

Steuersatz:	<ESC>**m** *d_1* [**;***d_2*] <CR>
Beispiel:	<ESC>m2<CR>

Parameter:
d_1 = Makro-Nr. (0),1,2 oder 3
d_2 = Funktion: 1 = Steuerparameter in *d_1* speichern, *d_1* wird zum Boot-Makro
Funktion: 2 = Boot-Makro (*d_1*) wechseln

Der Drucker besitzt 3 Makros, in denen die Werte aller Steuersequenzen fest abgespeichert werden können. Somit sind bereits beim Einschalten des Druckers die Geräteparameter bekannt.

Bitte beachten Sie, dass der Speichervorgang ca. 3 Sekunden benötigt und in dieser Zeit der Drucker nicht ausgeschaltet werden darf.

Boot-Makro 0 enthält die Grundeinstellung des Druckers und kann in Funktion 2 angewählt werden.
Wird der 2. Parameter nicht angegeben, gilt Funktion 1.

Die Grundeinstellung kann wieder hergestellt werden, wenn folgende Sequenzen an den Drucker gesendet werden:

<ESC>m0;2<CR>
mind. 5 Sekunden warten, dann Drucker aus und einschalten
<ESC>m3<CR>
mind. 5 Sekunden warten
<ESC>m2<CR>
mind. 5 Sekunden warten
<ESC>m1<CR>
mind. 5 Sekunden warten, dann Drucker aus und einschalten

Mit der Readback-Funktion kann her das aktuelle Makro das verwendet wird zurück gelesen werden.

Abrufen von gespeicherten Layout-Daten

Steuersatz:	<ESC>**y6**;<LAYOUT_NAME;><CR>
Beispiel:	<ESC>y6;STATI;<CR>

Der Drucker kann vordefinierte Layouts für verschiedene Etiketten intern abspeichern. Dieses ist sinnvoll, wenn z. B. immer wieder das gleiche Etikett ausgegeben werden soll, bei dem sich die Daten sich in der Regel nicht ändern.

Um nicht immer das gesamte Etikett übertragen zu müssen, kann mit dieser Steuersequenz das Etikett abgerufen werden.
Etwaige variable Datenfelder können vor Auslösung des Drucks übertragen werden.

Direkte Steuerbefehle

Steuersatz:	<ESC>**z***d*<CR>
Beispiel:	<ESC>z1<CR>

Die Steuersequenz ermöglicht Eingriffe in die Ablaufsteuerung des Druckers.
Dies macht jedoch nur Sinn , wenn dem Host-Rechner der aktuelle Zustand bekannt ist, d.h. dieser im direkten Dialog mit dem Drucker kommuniziert.

Schalter:

d = 0 Reaktiviert einen bereits stornierten oder abgearbeiteten Datensatz..(sofern noch im Drucker vorhanden).
Dabei wird das Statusbit 20 + 08 sowie ein Druckkommando mit der Anzahl 1 gesetzt.

d = 1 Führt einen Cutterschnitt durch.

d = 2 Wenn Etikettensynchronisation eingestellt ist, dann wird hier ein Formfeed durchgeführt, ansonsten wird das Papier 1,5 Zoll weiter transportiert und geschnitten.

d = 10 Alle Fehler der Kategorie **Warnung** werden angezeigt und müssen bestätigt werden.

d = 11 Alle Fehler der Kategorie **Warnung** werden angezeigt, müssen aber nicht bestätigt werden.

d = 12 Alle Fehler der Kategorie **Warnung** werden ignoriert.

d = 13 **Warnung #70** wird ignoriert.

ACHTUNG: Diese Steuerfunktion hat keine Readback-Funktion**.**

V24-Parameter

Steuersatz:	<ESC>**f**d_1**;**d_2**;**d_3**;***c*[**;0|1**]<CR>
Beispiel:	<ESC>19200;8;1;n<CR>

Parameter d_1:

9600	=	9600 Baud
19200	=	19200 Baud
38400	=	38400 Baud

Parameter d_2:

7	=	7 Datenbits
8	=	8 Datenbits

Parameter d_3:

1	=	1 Stopbits
2	=	2 Stopbits

Parameter c:

n	=	keine Parität
e	=	gerade Parität
o	=	ungerade Parität

Schalter:

0	=	kein XON/XOFF-Protokoll
1	=	mit XON/XOFF-Protokoll

Kapitel 8 : Sondersequenzen

Sondersequenzen sind nur für die werkseitige Einstellung des Druckers zu verwenden. Diese Sequenzen können grundlegende Einstellungen des Druckers ändern. Und die erforderliche Konfiguration des Druckers zerstören.

- ***Hinweis:***
  Dieses Sequenzen sind nur nach Rücksprache mit den Hersteller zu verwenden.

Setup-Konfiguration:

Steuersatz:	<ESC>**y30**;*d*<CR>
Beispiel:	<ESC>y30;80<CR>

Die Steuersequenz ermöglicht es das Setup-Konfigurationsbyte zu setzten. Um die Konfiguration permanent zu Speichern muss die Steuersequenz <ESC>y32<CR> ausgeführt werden.

Setup-Konfigurationsbyte:

Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
NU	FU4	FU3	FU2	FM	NU	NU	NU	FAB	NU	FU1	NU	NU	NU	FLT	NU

NU (Bit0) - Wird nicht benutzt.

FLT (Bit1) - Labeltaken erlaubt FLAG_LABEL_TAKEN

NU (Bit2) - Wird nicht benutzt.

NU (Bit3) - Wird nicht benutzt.

NU (Bit4) - Wird nicht benutzt.

FU1 (Bit5) - Applikationsspezifisch FLAG_OPTION_USER1 :
- Aktuell: keine Funktion

NU (Bit6) - Wird nicht benutzt.

FAB (Bit7) - Abschneider vorhanden (Defaulteinstellung) FLAG_ABSCHNEIDER

NU (Bit8) - Wird nicht benutzt.

NU (Bit9) - Wird nicht benutzt.

NU (Bit10) - Wird nicht benutzt.

FM (Bit11) - Schalter am Druckmodul vorhanden FLAG_MODUL

FU2 (Bit12) - Applikationsspezifisch FLAG_OPTION_USER2 :
- Aktuell: keine Funktion

FU3 (Bit13) - Applikationsspezifisch FLAG_OPTION_USER3 :
- Aktuell: keine Funktion

FU4 (Bit14) - Applikationsspezifisch FLAG_OPTION_USER4 :
- Aktuell: keine Funktion

NU (Bit15) - Wird nicht benutzt.

Setupdefaultwerte laden:

Steuersatz:	<ESC>**y31**<CR>

Die Steuersequenz ermöglicht es, dass die Setupdefaultwerte und Makrodefaultwerte neu geladen werden.

Setupdefaultwerte:

Boot-Makro = 0	(siehe Kapitel Makrosteuerung)
Setup-Konfigurationsbyte = 0x0080	(siehe Kapitel Setup-Konfigurationsbyte)
ABSTAND_LS_TK = 414 Stepps	Abstand zwischen Lichtschranke und Thermokopf.
ABSTAND_TK_AS = 264 Stepps	Abstand zwischen Thermokopf und Abschneider. Default Schneideposition.
DOTANZAHL_TK = 960 Dot	Dotanzahl Thermokopf
OFFSET_TK = 50 (50%-50=0%)	Offset für Thermokopfheizzeit.

Makrodefaultwerte:

Nachfolgende die wichtigsten Makrodefaultwerte:

Schneideposition	d = 0
Druckgeschwindigkeit	d= 125 mm/s
Thermokopfheizzeit d = 0	
Gerätedaten	d = 0100 (nach Reset 0180 Abschneider per default)
Erweiterte Gerätedaten	d = 0 (es sind keine erweiterten Gerätefunktionen aktiv)
Ländercode	d = 0 (USA)
Zeichensatz	d = 0 (ANSI)

Makrodefaultwerte laden:

Steuersatz: <ESC>**y32;*d***<CR>

Die mit der Steuersequenz <ESC>y30;d<CR> eingestellte Setup-Konfiguration wird abgespeichert.
Die im Makro abgelegten Parameter werden mit den aktuell eingestellten Werten geladen.

d -> 0 : Nur Setupmakros werden neu geladen.
d -> 1 : Setupmakros werden neu geladen Makros werde auf defaultwerte gesetzt.

Arbeitsspeicher neu Initialisieren:

Steuersatz: <ESC>**y33**<CR>

Die Steuersequenz ermöglicht es, dass der Arbeitsspeicher neu Initialisiert wird. Das Neuinitialisieren des Arbeitsspeichers findet dann beim nächsten Bootvorgang statt.
Die Betriebsdaten werden zurückgesetz

Kapitel 9 : Gerätespezifische Steuersequenzen

Es gibt die Möglichkeit, verschiedene Komponenten mittels Steuersequenz anzusprechen, konfigurieren und zu testen. Eine entsprechende Rückmeldung gibt Aufschluss über die Funktionalität der Komponente.

Initialisierung(<ESC>i..)

Steuersatz:	<ESC>**i** *d [|;d_1] [;d_4] [;d_3] [;d_4]*<CR>
Beispiel:	<ESC>i1|h125|s4|k30;1| k800;0|e0|t0<CR>

Mit dem Initialisierungsstring werden grundlegende Initialisierungen für die Einzelnen Schächte definiert.
Hierzu gehörten im wesentlichen die Definitionen des verwendeten Papiers.
Die Parameter werden durch ein ‚|' voneinander getrennt.
Der Initialisierungsstring muß für jeden Schacht einzel gesendet werden. Sinnvoll ist es wenn der Initialisierungsstring bei Inbetriebnahme des Systems gesendet wird.
Mit den Makrofunktionen kann man die Parameter permanent im Drucker speichern.
Die aktuelle Konfiguration kann mit der Sequenz ohne Angabe von Parameter angegeben werden. Z.B. für Schacht 1: <ESC>i1<CR>

Parameter:

d = Schacht für welchen die Initialisierungsprameter sind (1..4)

d_1, d_2...= Parmeter beginnen mit den jeweiligen Buchstaben:

k -> Druckerparameter , siehe (<ESC>k.. Parameter)

e -> Schneideposition , siehe (<ESC>e.. Parameter)

t -> TOF-Postion , siehe (<ESC>t.. Parameter)

w -> Heizzeitoffset , siehe (<ESC>w.. Parameter)

h -> Abstand der Synchronisationsmarken in mm (Bei Papier mit Synchronisation).
Hier ist zu beachten dass nur mit dem Parameter 10020;1 der Gerätedaten die Prüfung der Etikettenhöhe aktiviert wird.
Ist h=0 und 10020;1 aktiviert, dann wird keine Prüfung der Etikettenhöhe durchgeführt.

s -> Größe der Synchronisationsmarke in Druckrichtung in mm (Bei Papier mit Synchronisation).

Die Parameter können mehrfach (k-Parameter) und in einer beliebigen Reihenfolge angegeben werden .



Potiwerte (<ESC>p..)

Steuersatz:	<ESC>pd;d_1<CR>
Beispiel:	<ESC>p1;50<CR>

Parameter:
d = Lichtschranke (1 ..4)
d1 = Einstellwert (0..100%)

Die Steuersequenz legt den Strom für die einzelnen Lichtschanken fest. Es kann ein Stromwert von 0..100% vorgegeben werden. Wobei die Angabe nicht Linear ist.
Je nach verwendetem Papier können die Stromwerte unterschiedlich sein.

Lichtschranken:

d = 1 -> Synchronisationslichtschranke Durchlicht.
- Papierdetektion bei Schachtwechsel,
- Papierendedetektion.
- Synchronisation bei Durchlicht (<ESC>k20;1<CR>)

d = 2 -> Blackmarkdetektion
- Synchronisation bei Reflex (<ESC>k20;0<CR>)

d = 3 -> Blackmarkdetektion

d = 4 -> nu

Schachtsteuerfunktion (<ESC>s..)

Steuersatz:	<ESC>sd[;d_1]<CR>
Beispiel:	<ESC>s1<CR> <ESC>s1;i<CR>

Parameter:
d = Schacht (1 ..4)
d_1 = Zusatzfunktion (+;-;i)

Über diese Steuersequenz können verschiedene Aktivitäten auf die Papierschächte ausgeführt werden. Die Hauptfunktionalität ist die Durchführung des Schachtwechsels.

Wählt man einen Schacht an, bei dem das Papier in Parkposition steht, dann wird zuerst das Papier das sich in Druckposition befindet auf Parkposition gefahren. Nach dem das erfolgt ist, wird das Papier des angewählten Schachts auf Printposition gefahren.

Parameter d_1:

d_1 = ‚**i**' -> Abfrage Status Schacht. Es wird kein Schachtwechsel durchgeführt, vielmehr wird der Papierstatus für den im Parameter ‚d' angegebenen Schacht ausgegeben.

Papierstatus:

‚l' - Papier befindet sich in Parkposition

‚p' - Papier befindet sich in Druckposition

‚r' - Papier soll entnommen werden, ist aber noch nicht aus dem Schacht entfernt.

‚e' - Papierschacht ist Leer

‚?' - Keine Zuordnung möglich

d_1 = ‚**l**' -> Parken von Papier. Über diese Sequenz kann Papier eines beliebigen Schachts geparkt werden. Nach dem Parken steht der Drucker auf diesen Schacht.

d_1 = ‚**-**' -> Ausfädeln Papier. Das Papier wird aus dem angegebenen Schacht Parameter ‚d' ausgefädelt.
Diese Funktion ist sinnvoll, wenn Papier nur von einem Schacht entnommen werden soll.
Rücksetzen von Papierfehler (203, 160):
Mit dem Ausfädeln von Papier werden etwaige Fehlernummern 203 und die Information 160 zurück gesetzt. Ausschließlich mit dem Ausfädeln können die Fehlerzustände zurück gesetzt werden.
Wenn das Papier aus den Drucker entnommen werden soll, dann wird das Papier nach hinten herausgeschoben. Für diesen Schacht wird die Informationsnummer #160 (Papier entnommen) eingetragen. Problematisch ist nun das Fehlerhandling wenn das Papier nicht entnommen wird. Hier wird wie flogt vorgegangen:

- Papier wird nicht entnommen:
  Es ist nun nicht möglich auf einen anderen Schacht zu wechseln (Information #161 wird dann ausgegeben). Dieses liegt darin das der Schacht nicht geschlossen ist und somit das Papier noch lose im Schacht liegt.
- Papier wird geparkt mit <ESC>s[1..4]l<CR>
  Wenn das Papier geparkt wurde, dann kann normal auf die andere Schächte gewechselt werden.
  Für den betreffenden Schacht bleibt der Zustand Papierentnahme bestehen, da es nicht definiert ist ob das Papier bei einen etwaigen Feed eingezogen werden kann.
- Mit den Kommandos <ESC>s[1..4]-/+<CR> kann der Schacht normal wieder mit Papier bestückt werden.

d_1 = ‚**+**' -> Einfädeln Papier. Das Papier wird in den angegebenen Schacht (Parameter ‚d') eingefädelt.
Diese Funktion ist sinnvoll, wenn Papier in einen bestimmten Schacht bestückt wird.

Konfigurations- und Einstellfunktionen <ESC>y..

Der Drucker unterstützt eine Vielzahl von Parameter und Einstellungen. Die gängigsten Parameter werden über die Allgemeine-Steuersequenzen angesprochen. Werte, die sich nicht dauernd Ändern, werden über die Steuersequenz <ESC>y;.. angesprochen.

Um diese Werte und Einstellungen ändern und auslesen zu können wird die folgende Steuerstruktur verwendet.

Allgemeiner Datensatz Aufbau

Steuersatz:	<ESC>y**f;p[;p$_1$;p$_2$..]**<CR>
Beispiel:	<ESC>y6;1;50<CR>

Parameter:

1 – Monitor 1

1 – Uploader

2 – Monitor 2

6 – Flash Datensatz lesen.

9 – Testfunktionen.

11 – Papier/Ausweis Entwerten

12 – Einstellungen Feeder-Motoren.

13 – Einstellungen Sensoren Diodenströme usw.

15 – Darstellung Heizzeiten usw.

25 – Barcode Scanner.

30 – SETUP (Init Flash Setup).

31 – SETUP Defaultwerte lade.

32 – Makro Defaultwerte lade.

33 – Abspeichern von Setup und neu Initialisieren.

39 – Logwerte ausgeben.

41 – FPGA Status ausgeben.

41 – Antriebsmotor.

45 – Seriennummern.

Monitor 1:

<ESC>y1;kA7y<CR>

Die Monitor 1 Funktion dienet dazu dass aktuelle Programm zu verlassen und den Bootloader aufzurufen.

Über den Bootloader können dann eine neuen Fimware, FPGA, oder Konfigruationsdateien auf den Drucker aufgespielt werden.

Uploader:

<ESC>y1;kA7z<CR>

Der Uploader ist ein separates Programm innerhalb des Druckers. Er dient dazu die Firmware, FPGA kontrolliert auf den Drucker aufzuspielen.

Nach dem Aufruf der Steuersequenz wird der Uploader gestartet:

```
Updater PRINTON GmbH (c) 2006 / Uploader:V_xxxx
     1 - Firmware
     2 - FPGA
     3 - Font
     ? - Help
     m - Monitor
     g - go
```

Parameter :

- 1 : Firmware laden
- 2 : FPGA Laden
- 3 : Fontdatei laden
- ? : Ausgabe der Hilfe
- m : Monitor Starten
- g : Uploader verlassen und Applikation neu Starten.

Über die einzelnen Menüpunkte können die entsprechenden Programmmodule neu aufgespielt werden.

Damit das Programmmodul das Aufgespielt wird, auch auf Richtigkeit geprüft werden kann wird eine CRC16 Kontrollbyte verwendet:

Die CRC16 Bildung erfolgt nach folgenden Term :

$$x^{16} + x^{15} + x^2 + 1$$

Die Übertragung eines Programmmoduls beginnt immer mit der Angabe der Länge des Moduls (Angabe in Dezimal), dann die Checksumme (Angabe in Dezimal) und dann das eigentliche Modul in Binärform.

Nachdem die Übertragung vollständig abgeschlossen wurde, wird die Checksumme vom Uploader berechnet. Stimmt die berechnete Checksumme mit der zuvor übertragenen Checksumme überein, dann wird das Programmmodul geladen.

Ausgabe vom Uploader:

```
< Load Firmware

  Bitte Daten Senden !
     131072 (131072) Bytes
     CRC(Summ): 0x43
     CRC-16   : 0x2372 (0x2372)
+
OK
< OK
```

Monitor 2:

	<ESC>y2;s3rB<CR>
Beispiel:	<ESC>y2;<CR>

Wie Monitor 1 jedoch ohne aufspielen von neuen Firmware.

Flash-Datensatz lesen.:

	<ESC>y6;[MODU];<CR>
Beispiel:	<ESC>y6;Test1;<CR>

Über diesen Datensatz kann ein Modul (Steuersequenz für Fahrausweise) die im Flash des Druckers hinterlegt sind in den Eingabespeicher überspielt werden.

Testfunktionen:

	<ESC>y9;f;p<CR>
Beispiel:	<ESC>y9;1;<CR>

siehe Kapitel : Testfunktionen <ESC>y9;...

Papier/Ausweis entwerten (y11)

	<ESC>y11[;1]<CR>
Beispiel :	<ESC>y11<CR>

Manchmal ist es notwendig, einen bereits gedruckten Fahrausweis oder Reststück Papier zu entwerten, oder dass überschüssiges Papier ausgegeben wird.

Bereits gedruckte Fahrausweises dürfen zu diesem Zeitpunkt noch nicht geschnitten worden sein.

Beim Ausführen dieser Funktion wird das bedruckte oder überschüssige Papier zurücktransportiert und dann mit einem Entwertungsmuster neu bedruckt.

Sequenzen:

<ESC>y11<CR>	- Entwerten eines bedruckten Fahrausweises
<ESC>y11;1<CR>	- Entwerten eines Stück Restpapiers mit ausgeben des Reststücks (Maximal jedoch 20 cm).
<ESC>y11;2<CR>	- Prüfen ob eventuell überschüssiges Papier am Druckerausgang ansteht. Wenn ja, dann dieses Papier Entwerten und Schneiden.



Sensoren Diodenströme (y12)

	<ESC>y12;m;f;p<CR>
Beispiel:	<ESC>y12;1;4;30<CR>

Über dieses Steuersequenz können verschiedene Einstellungen für die Einzelnen Sensorstrecken vorgenommen werden.

Die Sensoren sind die Synchronisationssensoren und die Blackmarksensoren.

Auch kann über dieses Sequenz eine Kalibrierung der Sensoren vorgenommen werden.

Die Kalibrierungsfunktion ändert die Stromwerte des Sensorstroms kontinuierlich. Dabei wird geprüft wenn am Empfänger eine Signaländerung erfolgt. Dieser Wert wird dann ausgegeben. Das eingelegte Papier wird dann um ca. 1mm weitergesteppt und die Funktion wird dann erneut ausgeführt. Dieses wird 50 Mal wiederhlot.

Am Ende des Tests wird dann ein Vorschlag für die Einstellung des Sensorwerts ausgegeben.

Parameter :

```
m -> Auswahl des Sensors
     0 -> Parameter gilt für alle Sensoren
     1..4 -> Selektion Sensor 1 .. 4
f - > Auswahl der Art der Parameter
     0 - Sensor aus
     1 – Sensor ein.
     3 – Einstellung Grundfreqenz für PWM-Signal
          p – Grundfrequenz.
     4 – Einstellung Sensorstrom in %
         p – in % ( 0.. 100)
```

Die Diodenströme (<ESC>y12;m;4;..)können auch über die Steuersequenz <ESC>p..<CR> gesetzt werden.

```
     5 – Kalibrierungsfunktion wird für Sensor m
         ausgeführt (m kann nur 1..4 sein).
         Beispiel für Sensor 1:
            <ESC>y12;1;5<CR>
```



Feeder-Motoren (y13)

	<ESC>y13;m;f;p<CR>
Beispiel:	<ESC>y13;1;4;30<CR>

Über dieses Steuersequenz können verschiedene Einstellungen für die Einzelnen Feedermotoren vongenommen werden. Dieses sind Testfunktionen und Drehmomenteinstellungen.

Die Parameter für die Grundfrequenz und das maximale Drehmoment sind für alle Motoren gleich, da alle Motoren vom gleichen Typ sind und man davon ausgehen kann dass die Umgebungsbedingungen für jedem Motor identisch sind.

Parameter :

```
m -> Auswahl des Feeder Motors
     0 -> Parameter gilt für alle Motoren
     1..5 -> Selektion Motor 1 .. 5
f - > Auswahl der Art der Parameter
     0 – Motor aus
     1 – Motor linksherum (Timeout 1 Sek.)
     2 – Motor rechtsherum (Timeout 1 Sek.)
     3 – Einstellung Grundfreqenz für PWM-Signal
          p – Grundfrequenz.
     4 – Einstellung maximales Drehmoment über PWM.
          p – PWM in % ( 0.. 100)
     5 – Alle Feeder schließen wenn nicht mit Papier
         bestückt.
```

Auslesen Historybestrohmungswerte (y15)

<ESC>y15<CR>

Über diese Steuersequenz können die Historybestrohmungswerte ausgelesen werden.

Hier werden die Heizzeiten für die einzelnen Bestrohmungsphasen ausgegaben:

Ausgabe:

```
HZ1 (0xa258): 5960 477 µs
HZ2 (0xa256): 5410 433 µs
HZ3 (0xa254): 4810 385 µs
HZ4 (0xa252): 4100 328 µs
```

```
HZ5 (0xa250): 3680 294 µs
```

Bedeutung der einzelnen Zeitwerte:

HZ1 – Erstmalige Bestromung eines Dots (nur wenn die beiden vorrausgehenden Dots nicht bestromt wurde, wird T1 Benutzt).

HZ2 – Bestromung, wenn vorletztes Dot bestromt wurde jedoch das letzte nicht.

HZ3 – Bestromung, wenn vorletztes Dot bestromt wurde das letzte nicht und die benachbarten Dots bei der letzten Dotzeile bestromt wurden.

HZ4 – Wenn bei der letzten Dotzeile das Dot bestromt wurde.

HZ5 – Wenn bei den letzten beiden Dotzeilen das Dot bestromt wurde.

Barcode-Scanner (y25)

Steuersatz:	*<ESC>y25;d<CR>*
Beispiel:	*<ESC>y25;S0<CR> <ESC>y25;W1;i<CR>*

Diese Steuersequenz dient dazu, einen eingebauten Barcodescanner zu Konfigurieren.

Wird die Funktion ohne Parameter aufgerufen, dann wird der zuletzt gelesene Barcode ausgegeben.

Der Scanner wird im eingebauten Zustand für den PDM170 umkonfektioniert:

<ESC>y25;S0<CR>	Single Read Mode
<ESC> y25;A6<CR>	Liest nur Code 128
<ESC> y25;W1<CR>	Buzzer: Single Tone
<ESC> y25;W7<CR>	Buzzer: 50ms
<ESC> y25;T3<CR>	Buzzer: Loudness Minimum
<ESC> y25;Z2<CR>	Fest abspeichern

Die Ein- und Ausschaltung erfolgt im Programm über S7 (EIN) und S8 (AUS).

Einfacher Scanner-Test:

<ESC> y25;S2<CR>	Continous Read
<ESC> y25;S7<CR>	Scanner EIN



Motor (y41)

Steuersatz:	*<ESC>y41;d[;di]<CR>*
Beispiel:	*<ESC>y41;2;1500<CR> <ESC>y41;4;1500i<CR>*

Über dieser Steuersequenz kann der Motor angesteuert werden. Dabei mach der Drucker die Angegebene Schrittzahl nach vor oder zurück:

Parameter :

d :

- 1 – Motor vor in Vollschritten
- 2 – Motor vor in Halbschritten
- 3 – Motor zurück in Vollschritten
- 4 – Motor zurück in Habschritten
- 5 – Führt ein Form- Feed aus
- 6 – Auswerfen von Restpapier mit Entwertung maximal jedoch 20 cm Papier.

d_i : Anzahl der Schritte die ausgeführt werden sollen.

Serienummern (y45)

Steuersatz:	*<ESC>y45;d<CR>*
Beispiel:	*<ESC>y45;2<CR> <ESC>y45;0<CR>*

Über dieser Steuersequenz können die Verschiedenen Seriennummern der Komponenten ausgelesen werden, die Komponenten sind die Gesamtseriennummer des Druckers, die Seriennummer der Leiterkarte, Firmware informationen und FPGA info.

d :

- 0 – Alle Nummern und Informationen ausgeben
- 1 – Ausgabe der Gesamtseriennummer

  Ausgabe z.B::

  ```
  SerNr:1023 16.11.2006
  ```

- 2 – Ausgabe der Firmwareversion

  Ausgabe z.B::

  ```
  PDM170-V0.28/exp/S
  Mo 16.10.2006  8:12:47,51
  ```

- 3 – Ausgabe der Seriennummer der Leiterkarte

  Ausgabe z.B::

  ```
  LK: 27-000088
  ```

- 4 – Ausgabe der FPGA Informationen.

  Ausgabe z.B::

  ```
  FPGA Info:
  DeviceType: 2s100etq144
  CreateDate: 2006/08/15
  CreateTime: 11:02:02
  ```

Testfunktionen <ESC>y9;...

Allgemeiner Datensatz Aufbau

Steuersatz:	*<ESC>y9;f;p<CR>*
Beispiel:	*<ESC>y9;1;1<CR>*

Die Funktionen den innerhalb der Testfunktionen ausgeführt werden, dienen in der Regel ausschließlich Testzwecke.

Einzelne Funktionen sollen während des Betriebs nur mit Ausnahme verwendet werden.

Parameter:

f - *Funktion* (Dezimalwert)
p- *Parameter* (Dezimalwert)/ Parameter

Funktion :

Über die Funktion kann eine entsprechende Systemkomponente, die angesprochen oder getestet werden soll, ausgewählt werden.

Parameter:

Mittels der Angabe eines Parameters, wird eine entsprechende Aktion einer Komponente durchgeführt die unter Funktion ausgewählt wurde.

Funktionen:

2 – Cutter

5 – Sensoren-Abfragen

10 – Unterstütze Zeichensätze, Barcodes, Logos

11 – Ereignisspeicher (option)

12 – Testdruck

13 – Com-Paramter

14 – Betriebsdaten

16 - Temperatur Thermoleiste

20 - USB-Device

40 – Memory Dump I2C-Pro

Cutter (Funktion 2):

Beispiel:	*<ESC>y9;2;1<CR>*

Der Cutter kann hierdurch zu einem Schnitt veranlasst werden.

Tests:

1 – Einen Schnitt durchführen.

2 – Den Cutter 100ms Ansteuern.

Sensoren (Funktion 5):

Beispiel:	*<ESC>y9;5;0<CR>*

Hier wird der Zustand der Sensoren abgefragt.

```
LS41  -> nc
LS31  -> nc
LS20  -> 1=(Marke) Blackmark
LS10  -> 1=belegt Synchronisationssensor
LT0
SY0
PES0:1-> 1=belegt Papierende Sensor Schacht 1
PES1:1-> 1=belegt Papierende Sensor Schacht 2
PES2:1-> 1=belegt Papierende Sensor Schacht 3
PES3:1-> 1=belegt Papierende Sensor Schacht 4
PES4:0-> nicht benutzt
```

Unterstütze Zeichensätze, Barcodes, Logos (Funktion 10) :

Beispiel:	*<ESC>y9;10;0<CR>*

Mit Hilfe von dieser Funktion kann man sich die vom Drucker aktuell unterstützten Zeichensätze, Abgespeicherten Logos und unterstützten Barcodes ausgeben lassen.

Hierzu können verschiedene Parameter angegeben werden:

0 - Ausgabe der unterstützen Fonts

1 – Ausgabe der abgespeicherten Logos

2 – Ausgabe der unterstützten Barcodes

Testdruck (Funktion 12):

Beispiel:	*<ESC>y9;12;0<CR>*

Die Ausgabe des Testdrucks (Statusbelegs) wird auf dem Drucker initiiert.

0 – Ausgabe der Minimalinformationen.

1 – Ausgabe der Minimalinformationen.

2 – Ausgabe der intern abgespeicherten Fonts.

3 – Ausgabe eines Testmusters

4 – Ausgabe des Gerätedaten

Betriebsdaten (Funktion 14)

Beispiel:	*<ESC>y9;14;0<CR>*

Es werden die aktuellen Betriebsdaten angezeigt, auch können hier die Werte der Betriebsdaten zurück gesetzt werden.

Paramter:

0 – Anzeige der Betriebsdaten
Es werden mehrere Informationszeilen ausgegeben

0	->	Flag Bit 0 =>1 Papiervorendeloch erkannt.
1023	->	Zähler Cuts
304566	->	Bestromung Thermoleiste
13	->	Papierbewegung in Meter
125	->	Durchgeführte Papierwechsel

3 – Ausgabe des Papierverbrauchs.

Hier wird der Ermittelte Wert für das Papier das seit des letzten Einschaltens, bzw Rebboots gespendet wurde. Die Angabe erfolgt in Millimeter.

Messgenauigkeit::

Bei der Messung von Papier gibt es verschiedene Punkte welche zu einer Messungenauigkeit beitragen können. Diese Punkte sind:

- Schlupf Papier im Drucker
- Ungleichmäßiger Zug der Papierrolle
- Abnutzung/Ungenauigkeit Transportrollen



- Rundung bei Ausgabe
- Schachtwechsel
- „Motorfang“ (Nachdem Phasen abgeschaltet wurden)

Im Bezug auf obige Angaben können für die Messung einzelner Tickets eine Genauigkeit von ca.

+2mm / -(2mm – 3% Ticketlänge)

angegeben werden.

Ausgabeformat:

<ESC>y9;14;3;[SPENDELÄNGE]<CR>

Beispiel 550 Millimeter Papier wurden gespendet.

<ESC>y9;14;3;550<CR>

Zu Beachten ist, dass nach dem Einschalten ein kleiner Offset von ca. 28mm bis 32mm ausgegeben wird, wenn man den Papierverbrauch abfragt.

9 – Zurücksetzen der Betriebswerte auf Null

Temperatur Abfrage (Funktion 16)

Beispiel:	*<ESC>y9;16<CR>*

Mit dieser Steuersequenz kann die Temperatur des Thermokopfs und der Umgebung zurück gelesen werden.

Bei der Temperatur für den Thermokopf ist zu beachten, dass die Temperatur nicht der Temperatur der Thermozeile entspricht, sondern dass die ermittelte Temperatur des Thermokopfs entspricht.

Unter der Thermozeile werden die einzelnen Heizelemente verstanden. Der Thermokopf ist die Thermozeile inklusive der Montageplatte.

Als Rückgabewert wird die Temperatur in Grad Celsius zurück geliefert. Der Temperatur bereich reicht von ca. –15 Grad bis zu + 80 Grad.

Beim Rücklesen werden drei Parameter ausgegeben:

1. Parameter -> Temperatur Thermokopf
2. Parameter -> Temperatur Umgebung (Hier Leiterkarte)
3. Parameter -> nicht benutzt.

USB-Device (Funktion 20)

Beispiel:	*<ESC>y9;20;0<CR>*

Hier werden die grundlegenden Funktionalitäten des USB-Devices getestet. Es werden die Zustände der Flags abgefragt. Die USB-Leitung kann ein und Ausgeschaltet werden.

Parameter:

1 – Einschalten der USB-Leitung

2 – Abschalten der USB-Leitung

3 – Prüfe ob Daten um USB-Device stehen.

4 – Prüfe ob USB-Device für Daten bereit ist.

Memory Dump (Funktion 40)

Beispiel:	*<ESC>y9;40<CR>*

Ausgeben eines Memory Dumps des I2C-Proms

Kapitel 10 : Objektsequenzen

Aufbau:

Datensatz	= { Steuerblock, Layoutblock, Steuerblock }	
Layoutblock	= { <STX > Objektblock 1 [Objektblock *n*] <EOT> }	*n* = [2...]
Objektblock	= { [Objektsequenz *n*] Objekt }	*n* = [1...]
Objektsequenz	= { <ESC> *X* ... [<CR>] }	*X* = [A|C|...|Z]
Objekt	= { <ESC> **B**|**T**|**L**|**X** ... [<CR>] }	

Mittels Objektsequenzen wird ein Kartenlayout (Layoutblock) definiert.

Über mehrere Objektsequenzen kann ein Objekt in Form und Position bestimmt werden. Alle Objektsequenzen, die ein Objekt beschreiben, stellen einen Objektblock dar.
Die letzte Objektsequenz eines Objektblockes definiert den Objekttyp (Text, Barcode, Logo/Rahmen/Linie).

Die Anzahl der Objekte ist nicht begrenzt.

Alle Objektblöcke zusammen ergeben einen Layoutblock für 1 Etikett.

Auf den nachfolgenden Seiten sind die einzelnen Objektsequenzen detailliert beschrieben.

Y-Vergrößerung

Objektsequenz:	<ESC>**C***d* [<CR>]
	d = 1 ... 255: Y-Vergrößerungsfaktor
Beispiel:	<ESC>C2

Y-Faktor 1

Y-Faktor 2

Y-Faktor 3

Beispiel:

```
<STX>
<ESC>G50<ESC>I35<ESC>C1<ESC>TARIAL14f;Y-Faktor 1
<ESC>G50<ESC>I70<ESC>C2<ESC>TARIAL14f;Y-Faktor 2
<ESC>G50<ESC>I135<ESC>C3<ESC>TARIAL14f;Y-Faktor 3
<EOT>
```

X-Vergrößerung

Objektsequenz:	<ESC>**D***d* [<CR>]
	d = 1 ... 255: X-Vergrößerungsfaktor
Beispiel:	<ESC>D2

X-Faktor 1

X-Faktor 2

X-Faktor 3

Beispiel:

```
<STX>
<ESC>G50<ESC>I50<ESC>D1<ESC>TARIAL14f;X-Faktor 1
<ESC>G50<ESC>I90<ESC>D2<ESC>TARIAL14f;X-Faktor 2
<ESC>G50<ESC>I140<ESC>D3<ESC>TARIAL14f;X-Faktor 3
<EOT>
```

Zeichenabstand

Objektsequenz:	<ESC>**F***d* [<CR>]
	d = 1 ... 255: Anzahl Leerdots (=1/8mm)
Beispiel:	<ESC>F3

Es wird der Abstand zwischen 2 Zeichen festgelegt. Die hier definierten Leerdots werden zusätzlich zu den im Zeichenfont vorgegebenen Zeichenabstand eingefügt.

Zeichenabstand 1

Z e i c h e n a b s t a n d 2

Z e i c h e n a b s t a n d 3

Z e i c h e n a b s t a n d 4

Z e i c h e n a b s t a n d 5

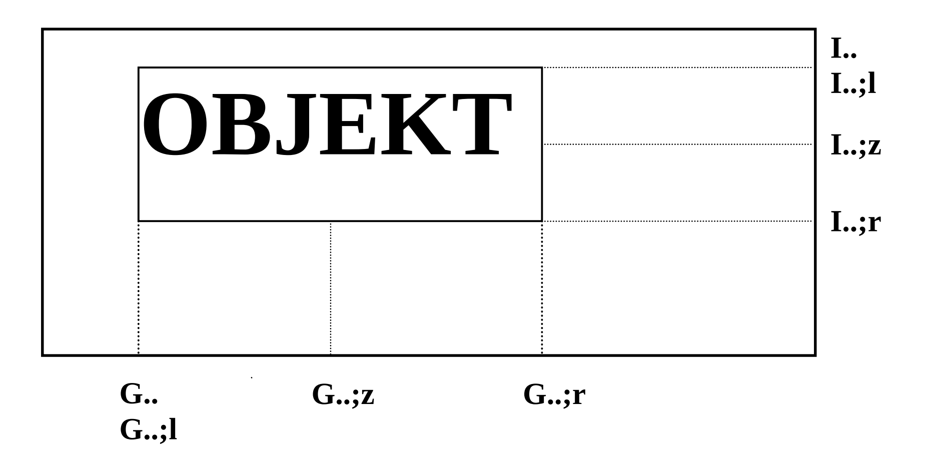
Beispiel:

```
<STX>
<ESC>G50<ESC>I50<ESC>F1<ESC>TARIAL14f;Zeichenabstand 1
<ESC>G50<ESC>I100<ESC>F2<ESC>TARIAL14f;Zeichenabstand 2
<ESC>G50<ESC>I150<ESC>F3<ESC>TARIAL14f;Zeichenabstand 3
<ESC>G50<ESC>I200<ESC>F4<ESC>TARIAL14f;Zeichenabstand 4
<ESC>G50<ESC>I250<ESC>F5<ESC>TARIAL14f;Zeichenabstand 5
<EOT>
```



Positionierung

Das Positionieren der Objekte erfolgt durch Angabe von X- und Y-Koordinate. Ohne weitere Angabe bezieht sich der so definierte Koordinatenpunkt auf die linke obere Ecke des betreffenden Text-, Barcode- oder Logo-Objektes. Durch zusätzliche Parameter kann als Bezugspunkt auch die Mitte oder der rechte- bzw. linke Rand festgelegt .werden. Hierdurch ist es möglich, sämtliche Objekte eines Drucklayouts sowohl in X- als auch in Y-Richtung zu zentrieren bzw. an beiden Rändern auszurichten.



X-Koordinate

Objektsequenz:	<ESC>**G***d* [**;** *x-ausrichtung*] [<CR>]	(X - Koordinate)
	d = 1 ... 832 :	Anzahl Dots vom linken Bildrand
	x-ausrichtung = l	linksbündig
	r	rechtsbündig
	z	zentriert
Beispiel:	<ESC>G10;z	

Y-Koordinate

Objektsequenz:	<ESC>**I***d..* [**;** *y-ausrichtung*] [<CR>]	(Y - Koordinate)
	d = 1 ... 1900 :	Anzahl Dots vom oberen Bildrand
	y-ausrichtung = l	linksbündig
	r	rechtsbündig
	z	zentriert
Beispiel:	<ESC>I10	

Fortschaltung (Option)

Objektsequenz:			<ESC>**Q**d_W;d_Z *[;d_F [;d_B [;d_A]]]* [<CR>]
d_W	=	-9 ... +9	: Fortschaltungswert
d_Z	=	1 ... 255	: Fortschaltungszyklus 1-254: Anzahl Etiketten ohne Fortschaltung 255 : Fortschaltung nach jedem Auftrag
d_F	=	<u>0</u> | 1	: Filter für führende Nullen (Textobjekte) 0 : führende Nullen drucken 1 : führende Nullen unterdrücken
d_B	=	<u>1</u> ...	: Beginn des Fortschaltungsfeldes Position der 1.Ziffer des Fort.feldes
d_A	=	<u>0</u> ...	: Feldgröße: Anzahl Ziffern = 0 : bis Ende Text-,Barcodestring : > 0 : Anzahl der relevanten Ziffern
Beispiel:	<ESC>Q1;1		

Die Fortschaltungsfunktion kann angewendet werden für **Text**- und **Barcode**-Objekte. Wird die Felddefinition so gewählt, dass das definierte Feld nicht komplett innerhalb des betreffenden Text- oder Barcodestrings liegt, so wird keine Fortschaltung getätigt.

Beispiele:

<ESC> Q 1 ; 1	nach jedem Etikett wird der komplette Textstring um +1 fortgeschaltet. Führende Nullen werden gedruckt.
<ESC> Q -1 ; 2 ; 1	nach jeweils 2 Etiketten wird der komplette Textstring um -1 fortgeschaltet. Führende Nullen werden durch Leerzeichen ersetzt. Ist der Wert des betreffenden Feldes = 0, so wird am Ende eine '0' gedruckt.
<ESC> Q 1 ; 1 ; 0 ; 4	nach jedem Etikett wird das relevante Ziffernfeld um +1 fortgeschaltet. Führende Nullen werden ausgedruckt. Das relevante Ziffernfeld beginnt mit der 4. Stelle des betreffenden Text-, Barcodestrings und endet am Stringende.
<ESC> Q 1 ; 1 ; 0 ; 4 ; 3	nach jedem Etikett wird das relevante Ziffernfeld um +1 fortgeschaltet. Führende Nullen werden ausgedruckt. Das relevante Ziffernfeld beginnt mit der 4. Stelle des betreffenden Text-, Barcodestrings und erstreckt sich über insgesamt 3 Ziffern.

Drehung

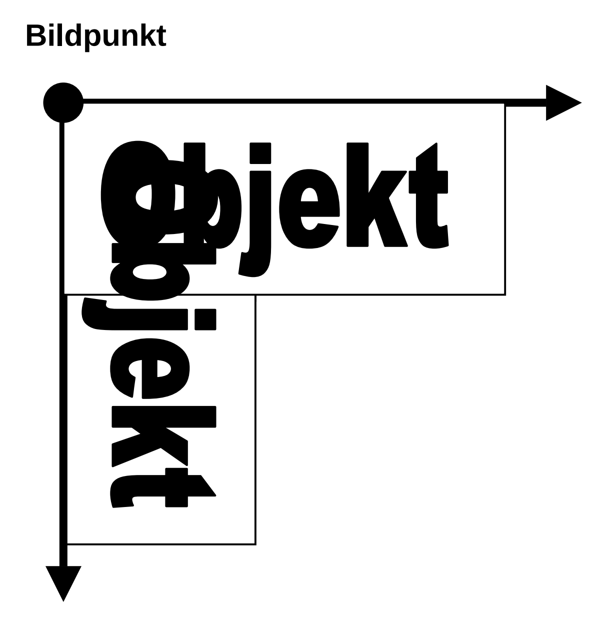
Objektsequenz:	<ESC>**R <u>0</u>**|**90**|**180**|**270**[<CR>]
Beispiel:	<ESC>R0

Jedes Objekt kann um 90°, 180° oder 270° gedreht werden.

Beispiel: Drehung des Objektes ohne Ausrichtung.
Dabei wandert der Bezugspunkt am Objekt gegenüber dem festen Bildpunkt.

```
<STX>
<ESC>G100<ESC>I200<ESC>R0<ESC>TArial12f;Objekt
<ESC>G100<ESC>I200<ESC>R90<ESC>TArial12f;Objekt
<ESC>G100<ESC>I200<ESC>R180<ESC>TArial12f;Objekt
<ESC>G100<ESC>I200<ESC>R270<ESC>TArial12f;Objekt
<EOT>
```

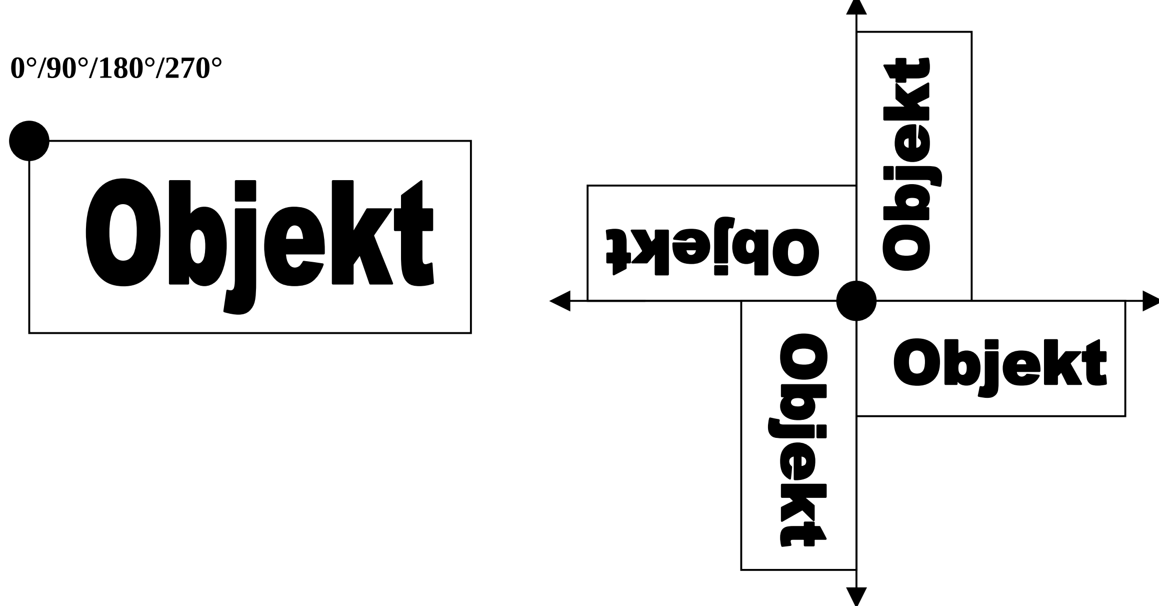






Beispiel: Drehung des Objektes mit Ausrichtung.
Dabei bleiben der Bezugspunkt am Objekt sowie der Bildpunkt fest.

```
<STX>
<ESC>G100<ESC>I200<ESC>R0<ESC>TArial12f;Objekt
<ESC>G100;r<ESC>I200<ESC>R90<ESC>TArial12f;Objekt
<ESC>G100;r<ESC>I200;r<ESC>R180<ESC>TArial12f;Objekt
<ESC>G100<ESC>I200;r<ESC>R270<ESC>TArial12f;Objekt
<EOT>
```



Fettdruck

Objektsequenz:	<ESC>**H**d [CR>]
	d = 1 – 4 (Dots)
Beispiel:	<ESC>H1

Texte können fett gedruckt werden.
Über den Parameter d kann eingestellt werden, um wie viele Dots der Text stärker gedruckt werden soll.

Beispiel einen Text 2 Dot stärker drucken.

```
<STX>
<ESC>G100<ESC>I200<ESC>H2<ESC>TArial12f;Objekt
<EOT>
```

Underline

Objektsequenz:	<ESC>**U**d [CR>]
	d = 1 – 2 (Anzahl Linien)
Beispiel:	<ESC>U1

Mit Hilfe dieser Objektsequenz, können Texte unterstrichen gedruckt werden. Der Parameter d gibt die Anzahl der Linien unter dem Text an (max. 2). Wird kein Parameter angegeben wird automatisch eine Linie unter dem Text gedruckt.

Beispiel für einen doppelt unterstrichenen Text:

```
<STX>
<ESC>G100<ESC>I200<ESC>U2<ESC>TArial12f;Objekt
<EOT>
```

Variable Objekte

Objektsequenz:	<ESC>**V**a_{obj} [CR>]
	a_{obj} = Objekt-Name : 0-9 , A-Z , a-z (1 Zeichen)
Beispiel:	<ESC>V1

Jedes Objekt kann als variabel definiert werden. Dies eröffnet die Möglichkeit, außerhalb des Layoutblocks diesem Objekt einen neuen Inhalt zuzuweisen, ohne dass das komplette Etikett neu übertragen werden muss.

Das im Layoutblock definierte Originalobjekt gibt die maximale Feldgröße vor. Später kann dieses Objekt maximal mit der gleichen Datenmenge gefüllt werden.

Die Übergabe neuer Daten für Text- und Barcodefelder erfolgt mit der Steuersequenz <ESC>va_{obj} , für Logos mit <ESC>la_{obj}.



Bildhintergrund

Objektsequenz: <ESC>**YZ;***HH*$_{Bilddaten}$<CR>

$Z_{Zeilennummer}$	Zeilennummer die Übertragen wurde
$HH_{Bilddaten}$	Bilddaten in Binärform Jedes Bit eines Bytes repräsentiert 1 Punkt Bit=0: Punkt nicht drucken, =1: Punkt drucken

Datenbit:	7	6	5	4	3	2	1	0
Punkt:	1	2	3	4	5	6	7	8

Normalerweise ist der Bildhintergrund bzw. Bildspeicher gelöscht (weiß). Mit Hilfe dieser Sequenz kann man den Bildhintergrund beschreiben. Es handelt sich hierbei um keine echte Objektsequenz, da die Bildsequenzen direkt im Bildspeicher landen und somit Objektattribute nicht anwendbar sind. Dafür kann aber der gesamte Bildbereich ohne 64K-Begrenzung beschrieben werden.

Bei sofortiger Bearbeitung des Datensatzes muss im Eingabespeicher nur 1 Bildzeile zwischengespeichert werden.

Während Objekte normalerweise in den Bildbereich geodert werden, überschreiben Bildsequenzen den Bildspeicher. Aus diesem Grund sind Bildsequenzen vor Objekte an den Drucker zu übertragen.

Je Bildzeile wird 1 ESC-Sequenz übertragen. Danach wird automatisch die nächste Bildzeile selektiert. Die max. Anzahl Bildsequenzen ist mit der Bildhöhe identisch. Zuviel Bildsequenzen werden mit einer WARNUNG-Meldung ignoriert.

Die Anzahl Bilddaten je Sequenz errechnet sich aus der Bildbreite / 8. Kommastellen werden aufgerundet. Falsche Anzahl Bilddaten führen zu unvorhersehbaren Fehler und sollten nach Möglichkeit vermieden werden.

Zur Reduzierung der Datenmenge können Bildzeilen, in denen nichts gedruckt wird, durch <ESC>**Z***d* übersprungen werden. Der Parameter *d* gibt die Anzahl leerer Bildzeilen an.

Beispiel:

Etikettenbreite = 800 Dots:	Je Bildzeile sind 100 Bilddaten zu senden.
Etikettenhöhe = 480 Dots:	Es dürfen max. 480 Bild-Sequenzen zum Drucker geschickt werden.



Kapitel 11 : Modulverwaltung

Modulverwaltung bearbeiten

Steuersatz:	<ESC> **>** x_1;x_2[;**s**]<CR>
Beispiel:	<ESC> > d;d;logo3;<CR>

```
<ESC> > x1 ; x2 [;s] <CR>
        f -------------> Programm-Flash-Zugriff
            i --------->   Modulname abfragen
        d -------------> Daten-Flash-Zugriff
            0 --------->   Modul-Downloading
            1;modul; -->   Modul-Uploading
            1;* ------->   Daten-Flash-Uploading
            2;modul; -->   Modul-Downloading Auto d.h.
                             die Modulgrösse wird mit dem
                             Upload von der
                             Modulverwaltung ermittelt
            i --------->   Modulname abfragen
            b --------->   verfügbare Speichergröße
                             abfragen
            c --------->   freie Speichergröße
                             abfragen
            m --------->   Anzahl der Module im Daten-
                             Flash und Programm-Flash
                             abfragen.
            d;modul;--->   Löschen eines Moduls im
                             Verzeichnis
            d;*-------->   Löschen des gesamten Daten-
                             Flash
            d;*;j------->  Löschen des gesamten
                             Daten-Flash ohne Rückfrage
            r --------->   Defragmentieren
```

Über diese Steuersequenz wird die gesamte Modul-Verwaltung abgewickelt. Sie beinhaltet das Down- und Uploading sowie das Löschen von Modulen. Innerhalb des Daten-Flashs können einzelne Module ergänzt oder gelöscht werden.
Gut-Quittierungen enden mit '+', Schlecht-Quittierungen mit '-'.

Die Modulverwaltung kann insgesamt **64** (Standardkonfiguration) **Module** verwalten.
Somit hängt die Anzahl der Module, welche noch in das Datenflash geladen werden können, von der Anzahl der bereits im Programmflash hinterlegten Module ab.

Rückmeldungen :

```
Rückmeldung + :
Immer dann, wenn ein Modul erfolgreich abgespeichert werden
konnte und bei Defragmentierung, wenn die Defragmentierung
funktioniert hat.

Rückmeldung - :
-> Bei Fehlerhafter Defragmentierung.
-> Wenn der zur Verfügung stehende Speicherbereich
   durch das Modul überschritten wird.
-> Wenn die Headerkennung "KeNuUnG" eines Moduls
   das geladen werden soll nicht Stimmt.
-> Wenn das Modul nicht an einem Stück in den
   Speicher geschreiben werden kann.
-> Wenn nicht das komplette Modul in den Speicher
   geschrieben werden konnte.
-> Beim Löschen eines Moduls: Das abschließende ';'
   nach dem Namen in der Esc-Sequenz fehlt.
-> Beim Löschen eines Moduls: Das Modul mit dem angebebenen
   Namen konnte nicht im Speicher gefunden werden.
-> Löschen des Flashs war nicht erfolgreich.
```

Zugriffe auf das Programm-Flash:
64kbyte Programm-Flash zum Abspeichern von z.B. Fonts, Barcodes.

Hinweise zu f;i :
Ausgabe aller im Programm-Flash befindlichen Modulnamen:

Bsp.: `B:EAN8;B:EAN13;F:ARIAL08F;F:ARIAL09F;...<CR><LF>`

Zugriffe auf das Daten-Flash :
In diesem 64KB Flash können Module langfristig abgespeichert werden. Es können einzelne Module ergänzt oder gelöscht werden.

Hinweise zu d;0 :
Eine Datei darf nur 1 Modul beinhalten und benötigt als Abbruchkriterium am Ende **EOF.**
Bsp.: `<ESC>>d;0<CR><Modul><$EOF$>`

Hinweise zu d;1;modul; :
Rücklesen eines Moduls über die serielle Schnittstelle.
Es können auch Module außerhalb des Daten-Flashs selektiert werden.

Hinweise zu d;1;* :
Rücklesen aller Module innerhalb des Daten-Flashs über die serielle Schnittstelle.

Hinweis zu d;2;modul;:
Es wird zuerst der Modulheader an die Modulverwaltung übertragen. Der Modulheader hat jedoch noch keine definierte Modullänge. Die tatsächliche Modullänge wird erst mit derm Übertragen der eigentlichen Moduldaten ermittelt. Die wird z.B. beim Download der PCX-Dateien innerhalb der Emulation verwendet.

Hinweise zu d;i :
Ausgabe aller im Daten-Flash befindlichen Modulnamen.

Bsp.: `F:LUCIDA22F;F:GAMOND14F;W:LOGO1;...<CR><LF>`

Hinweise zu d;b :
Der angezeigte freie Speicherplatz bezieht sich immer auf den momentan verfügbaren Bereich.
Bsp: 434200<CR>

Hinweise zu d;c :
Der angezeigte freie Speicherplatz bezieht sich auf den möglichen freien Bereich nach einer Defragmentierung.
Bsp: 498480<CR>

Hinweise zu d;m :
Es wird die max. Anzahle der möglichen Module (Module im Programmflash + Module im Datenflash) angezeigt, sowie die Anzahl der Module im jeweiligen Flashbereich.

Hinweise zu d;d;modul; :
Es wird der angegebene Modulname gelöscht und das Verzeichnis neu aktualisiert. Der Speicherbereich wird dadurch noch nicht freigegeben.

Hinweise zu d;d;* :
Es wird der Daten-Flash gelöscht und das Verzeichnis neu aktualisiert. Danach steht der gesamte Speicherbereich zur Verfügung.

Hinweis zu d;d;*;j
Wie d;d;* löschen des gesamten Daten-Flash, jedoch ohne Rückfrage.

Hinweise zu d;r :
Reicht der angezeigte freie Speicherplatz nicht mehr aus, muss defragmentiert werden. Dies ist jedoch nur sinnvoll, wenn sich gelöschte Module im Daten-Flash befinden. Den dabei entstehenden Speicherplatz kann man vorab mit <ESC>d;c<CR> abfragen.
Der Vorgang kann einige Minuten dauern.
Es findet intern keine Flash-Programmierung statt, wenn es nichts zu defragmentieren gibt.
Beendet wird der Vorgang mit '+' oder '-' – Quittierung

Kapitel 12 : Aufbau eines Moduls

Mit Hilfe der Modulverwaltung können nur Dateien mit einem bestimmten Format im Programm- bzw. Datenflash des Druckers hinterlegt werden.

Modulheader

Ein Modul kann Font-, BMP- oder eine Etikettdatei beinhalten.
Jedes Modul benötigt zur Identifikation seinen eigenen Header (40 Bytes) mit folgendem Aufbau:

1. Kennung (0x00):	#KeNnUnG	(8 Byte)
2. Identifikation (0x08):		(1 Wort, Hexformat)
	0002	(Font)
	0004	(Logo)
	0008	(Datensatz,Etikett)
	0010	(BMP)
	0020	(IniFile)
3. Typ (0x0a):	FF	(1 Byte, Hexformat)
4. Reserve(0x0b):	0	(1 Byte)
5. Modulname (0x0c):	....	(16 Zeichen) Bei Modulnamen < 16 Zeichen sind die restlichen Stellen mit Punkten '.' aufzufüllen.
6. Reserviert (0x1c):	(4 Bytes)	
7. Modullänge (0x20):		(4 Bytes) Gesamtlänge Modul (LSB,MSB)
8. Reserviert (0x24):	(4 Bytes)	
9. StartDaten (0x28):		Ab hier stehen die Font-, Barcode oder Etikettendaten.

Die Datei wird im Anschluß nach der Steuersequenz gesendet und endet mit einem TimeOut von ca. 2 Sekunden.

Modulende

Ein Modul muss immer mit einem **$**-Zeichen enden.



Kapitel 13 : Objekte

Ein Objekt wird über eine spezielle Objektsequenz definiert. Diese Angabe muss immer als letzte Objektsequenz in einem Objektblockes stehen. Die Angaben in den voranliegenden Objektsequenzen beziehen sich auf dieses Objekt.

Der Drucker unterscheidet folgende Objekte:

- <E_C> T ... TEXT-Objekt
- <E_C> B ... BARCODE-Objekt
- <E_C> L ... LOGO-Objekt
- <E_C> M ... gespeichertes LOGO-Objekt im Flash
- <E_C> X ... LINIE-/RAHMEN-Objekt
- <E_C> S ... GERADE-/LINIE-Objekt

Objekte vom Typ Text, Barcode und Logo werden mit <ESC>V*a* *als* variabel gekennzeichnet.

Variable Objekte werden mit neuen Daten aufbereitet über

- <E_C> v ... TEXT-Objekt
- <E_C> v ... BARCODE-Objekt
- <E_C> l ... LOGO-Objekt

Text

Objektsequenz: <ESC>**T***fonttyp***;***textdaten* [<CR>]

Beispiel Fonttypen:

fonttyp	Textbeispiele:
COURIER08f	**Courier 08 fett**
COURIER10f	**Courier 10 fett**
COURIER12f	**Courier 12 fett**
COURIER14f	**Courier 14 fett**
ARIAL08f	**Arial 08 fett**
ARIAL09f	**Arial 09 fett**
ARIAL10f	**Arial 10 fett**
ARIAL12f	**Arial 12 fett**
ARIAL14f	**Arial 14 fett**
ARIAL16f	**Arial 16 fett**
ARIAL18f	**Arial 18 fett**

Text-Objekte werden definiert durch Angabe des *Fonttyps* und des betreffenden Zeichenstrings. Bei falschem oder fehlendem *Fonttyp* wird als Ersatztyp Courier 08 fett selektiert.
Der Fontname kann bis zu 16 Stellen beanspruchen. Es wird hier nicht auf Groß-/Kleinschreibung geprüft.

Über die Modulverwaltung können noch weitere Fonts in den Drucker geladen werden. Es ist auch möglich mit Hilfe der Modulverwaltung Fonts zu entfernen.

1. Beispiel: festes Textfeld

```
<STX>
<ESC>TArial6f;Textstring
<EOT>
<ESC>#1<CR>
```

2. Beispiel: variablesTextfeld

```
<STX>
<ESC>Vn
<ESC>TArial6f;Schmitt
<EOT>
<ESC>#1<CR>
<ESC>vn;Maier
<ESC>#1<CR>
```

Logo

Objektsequenz: <E_C> **L** *d..*$_{\text{Breite}}$ **;** *d..*$_{\text{Höhe}}$ **; l ;** *HH*$_{\text{logodaten}}$ <C_R >

d..$_{\text{Breite}}$:	Logo-Breite : Anzahl Dots in X-Richtung
d..$_{\text{Höhe}}$:	Logo-Höhe : Anzahl Dots in Y-Richtung
l	Logotyp (kleines L)
HH$_{\text{logodaten}}$	Logodaten in Binärform Jedes Bit eines Bytes repräsentiert 1 Punkt Bit=0: Punkt nicht drucken, =1: Punkt drucken Datenbit: 7 6 5 4 3 2 1 0 Punkt: 1 2 3 4 5 6 7 8
<C_R>	muss direkt hinter den Logodaten stehen !!

Als Logo wird ein frei programmierbares Bild verstanden, dessen Punktmuster (Bildmap) dem Drucker als Daten übergeben werden. Ein Logo-Objekt kann wie andere Objekte auch durch die entsprechenden Objektsequenzen in Form und Position näher festgelegt werden.

1. Beispiel: festes Logo

```
Logo:        Bit:  7654 3210    Hexwert   dezimal

....¦...           0000 1000      08          8
....¦...           0000 1000      08          8
¦¦..¦...           1100 1000      C8        200
..¦.¦...           0010 1000      28         40
...¦....           0001 0000      20         32
```

```
<STX>
<ESC>G50<ESC>I35<ESC>R0<ESC>C4<ESC>D4
<ESC>L8;5;l;08 08 C8 28 20<CR>
<EOT>
<ESC>#1<CR>
```

2. Beispiel: variables Logo

```
<STX>
<ESC>G50<ESC>I35<ESC>R0<ESC>C4<ESC>D4
<ESC>Vg
<ESC>L8;5;l;08 08 C8 28 20 <CR>
<EOT>
<ESC>#1<CR>
<ESC>lg;8;5;ff ff ff ff ff <CR>
<ESC>#1<CR>
```



Flash - Logo

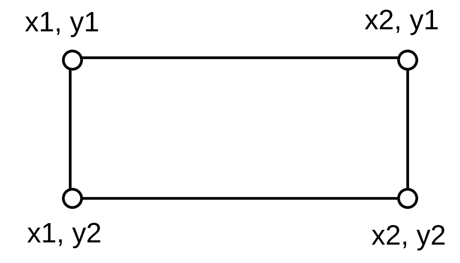
Objektsequenz:	<ESC>**M** *Logoname* **;** [<CR>]
Beispiel:	<ESC>MFirmenlogo;

Das Logo liegt bereits im Flash-Speicher vor und kann unter dessen Namen angesprochen werden. Es lässt sich wie jedes Objekt behandeln.
Der Logoname darf max. 16-stellig sein.

Linie und Rahmen

Objektsequenz:	<ESC>**X**d_{X1}**;**d_{Y1}**;**d_{X2} **;**d_{Y2}**;**d_{Breite} [<CR >]	
	d_{X1} , d_{Y1} :	Koordinate 1 : 1 ... Etikettbreite
	d_{X2} , d_{Y2} :	Koordinate 2 : 1 ... Etiketthöhe
	d_{Breite} :	Linienbreite: Anzahl Dots : 1 ...

Über diese Objektsequenz können sowohl Linien als auch Rahmen erzeugt werden. Falls [x1,y1] und [x2,y2] eine diagonale Linie beschreiben, wird ein Rahmen gedruckt, ansonsten eine waagerechte oder senkrechte Linie.

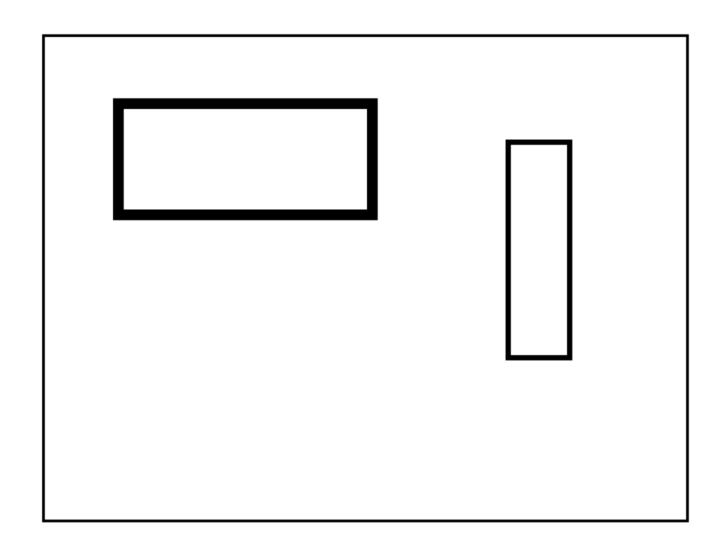


Es ist zu beachten, dass das 2. Koordinatenpaar positiver sein muss als das erste.

Linien werden nach innen verbreitert.

Beispiel:

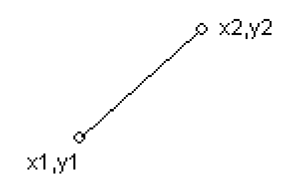
<STX>
<ESC>X20;20;250;150;6
<ESC>X300;40;350;330;3
<EOT>



Gerade und Linie

Objektsequenz:	<ESC>Sd_{X1};d_{Y1};d_{X2};d_{Y2};d_{Breite} [<CR >]
	d_{X1}, d_{Y1} : Koordinate 1 : 1 ... Etikettbreite
	d_{X2}, d_{Y2} : Koordinate 2 : 1 ... Etiketthöhe
	d_{Breite} : Linienbreite: Anzahl Dots : 1 ...

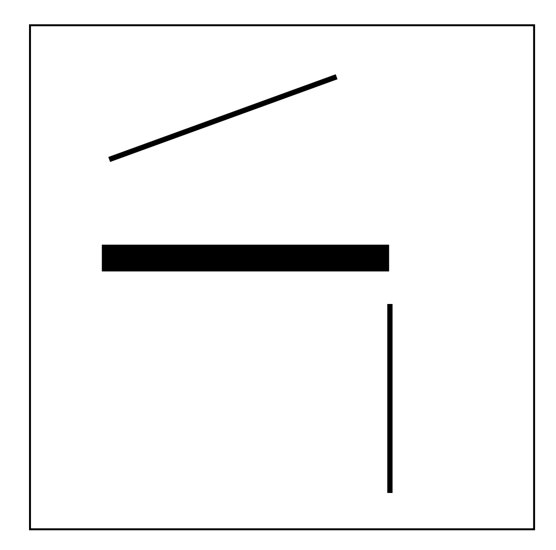
Über diese Objektsequenz können senkrechte, waagrechte sowie diagonale Linien erzeugt werden.





Beispiel:

<STX>
<ESC>S600;750;600;450;2
<ESC>S150;400;600;400;10
<ESC>S150;250;500;125;2
<EOT>

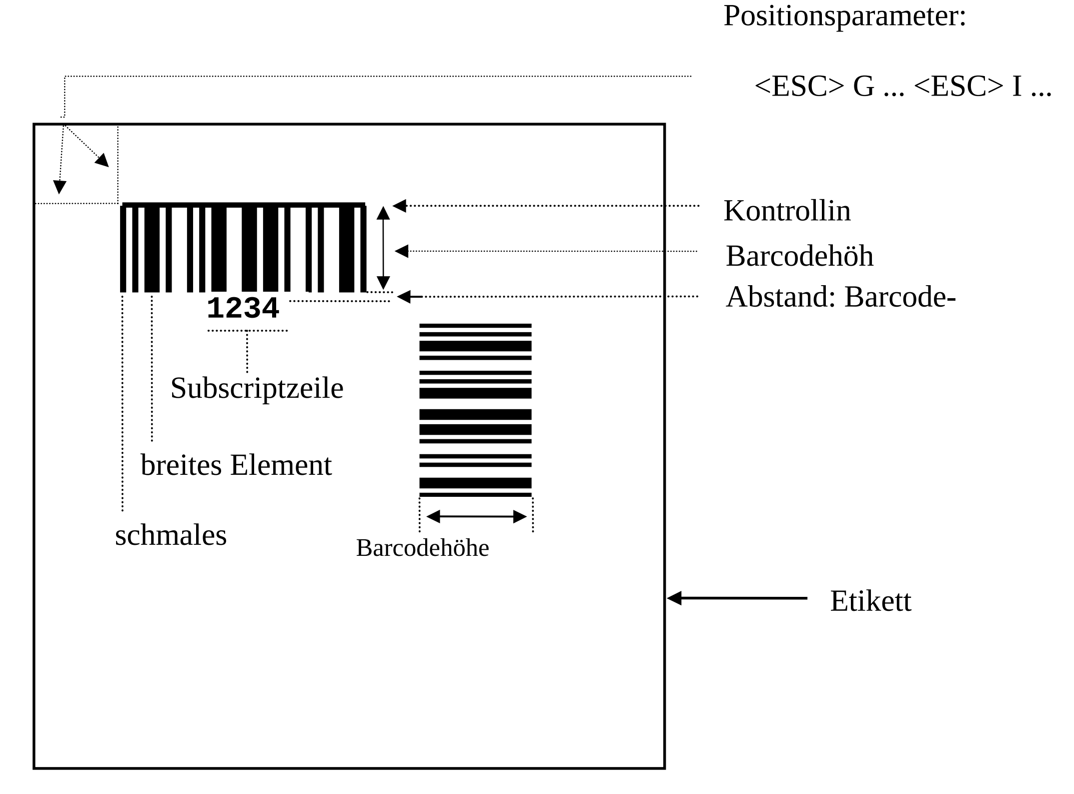




Barcode – Objekt (Optional nur auf Anfrage)

Einführung

Barcode-Objekte können durch eine Vielzahl an Parameter in Form und Funktion gestaltet werden. In der nachfolgenden Skizze sind diese Parameter veranschaulicht.



Die Einbindung eines Barcodes erfolgt über folgende Objektsequenz:

<ESC> **B** *S*$_{Typ}$ **;** [*parameter* **;** ...] > *barcodedaten* [CR]

S_{Typ} = Name des Barcodetyps :

S_{Typ}	entsprechender Barcode:
C_25_I	Code Interleaved 2 of 5
C_39	Code 39
C_128	Code 128
EAN8	EAN-8
EAN13	EAN-13
EAN128	EAN-128
PDF417	PDF-417

Als erster Parameter muss immer der gewünschte Barcodetyp eingegeben werden. Danach können weitere optionale Parameter folgen. Jede Parametereingabe beginnt mit einem Kennbuchstaben und wird mit einem Semikolon ' **;** ' abgeschlossen. Der Beginn der Barcodedaten wird mit dem Zeichen ' **>** ' angezeigt.

Optionale Parameter:

Hier können weitere barcodespezifische Parameter vorgegeben werden.
Bei Nichteingabe wird ein Ersatzwert angenommen.
Die Reihenfolge der Parametereingabe (Barcode, Subscript) ist beliebig.
Die Angaben in den Parametern sind entweder als Dezimalzahl oder als Textstring zu tätigen. Nachfolgend eine Liste der optionalen Parameter:

Parameter für Barcodeteil:

- **H** *d* — **Höhe** des Barcodes in Anzahl Dots (1/8 mm bzw. 1/12 mm)

  Ersatzwert: H120

- **B** *d* — **Breite** für ein schmales Element in Anzahl Dots
  Ersatzwert: B3

- **R 2 | 3 | 5** — **Ratio** = Verhältnis von breitem Element zu schmalem Element

  2 = 2 : 1
  3 = 3 : 1
  5 = 5 : 2
  Ersatzwert: R3

- **K 0 | 1** — **Kontrollinie** : 0 = OHNE , 1 = MIT Kontrollinie
  Ersatzwert: K0

- **Z 0 | 1 | 2** — **Prüfziffer** : 0 = OHNE , 1 | 2 = MIT Prüfziffer
  Z1 : Prüfziffer erscheint nicht in der Subscriptzeile
  Z2 : Prüfziffer wird auch in der Subscriptzeile ausgedruckt

  Bei einigen Barcodes kann optional eine zusätzliche Prüfziffer generiert werden. Barcodes, in denen die Prüfziffer fester Bestandteil ist, sind hiervon nicht betroffen.
  Ersatzwert: Z0

- **S 0 | a | b | c** — **Startcode bei Code 128**
  Ersatzwert: S0

Parameter für Subscriptzeile:

Für jeden Barcode kann automatisch eine Subscriptzeile ausgedruckt werden. Diese wird mittig unter den Barcode gedruckt. Über folgende Parameter kann die Subscriptzeile weiter gestaltet werden:

- **A** *d* — **Subscriptzeile: Objekt-Attribute**
  Ersatzwert: A0

- **T** *fonttyp* — **Subscriptzeile: Zeichenfont**
  Es können alle installierten Fonts benutzt werden.
  Ersatzwert: COURIER08f

- **C** *d* **Subscriptzeile: Y-Faktor**
  Ersatzwert: C1
- **D** *d* **Subscriptzeile: X-Faktor**
  Ersatzwert: D1
- **F** *d* **Subscriptzeile: Zeichenabstand** (Anzahl Dots)
  Ersatzwert: F1
- **P** *d* **Subscriptzeile: Abstand zum Barcode** (Anzahl Dots)
- **P %** Bezugspunkt ist die letzte Dotlinie des Barcodesymbols. P0 würde die Subscriptzeile bündig unter das Barcodesymbol drucken.

  Ein negativer Wert plaziert die Subscriptzeile in das Barcodesymbol hinein, positive Werte entfernen die Subscriptzeile vom Barcode.
  P % : **Unterdrückung der Subscriptzeile.**
  Ersatzwert: P1

Weiterhin können zur Gestaltung und Plazierung des Barcodesymbols auch folgende Objektsequenzen verwendet werden:

- <ESC>A ... Objekt-Attribute
- <ESC>G ... X-Position
- <ESC>I ... Y-Position
- <ESC>Q ... numerische Fortschaltung
- <ESC>R ... Drehung
- <ESC>V... variables Objekt

Beispiel: Etikett mit variablen Code 39 - Feld

```
<STX>
<ESC>V1
<ESC>G50<ESC>I40<ESC>R0
<ESC>BC_39;H70;K1;B3;R2;Z1;TARIAL20f;F2;P1;>HEIDELBERG
<EOT>
<ESC>v1;MANNHEIM<CR>
<ESC>#1<CR>
```



Code 2 of 5 Interleaved (Optional)

Objektsequenz: <ESC>**BC_2o5_I;** [*parameter* **;** ...] **>** *barcodedaten* [<CR>]

Unterstützung optional

parameter :	**Barcode:**
H 1 ... 1000	Barcodehöhe in Anzahl Dots Ersatzwert: H120
B 1 ... 99	Breite eines schmalen Elementes in Anzahl Dots Ersatzwert: B3
R 2 | 3 | 5	Ratio = Verhältnis von breitem Element zu schmalem Element 2 = 2 : 1 , 3 = 3 : 1 , 5 = 5 : 2 Ersatzwert: R3
K 0 | 1	Kontrollinie: 0 = OHNE , 1 = MIT Kontrollinie Ersatzwert: K0
Z 0 | 1 | 2	Generierung bzw. Ausdruck des Prüfzeichens in der Subscriptzeile 0 = kein Prüfzeichen generieren 1 = Prüfzeichen im Barcode, aber nicht in der Subscriptzeile 2 = Prüfzeichen im Barcode und in der Subscriptzeile Ersatzwert: Z0
	Subscriptzeile:
A *d*	Objekt-Attribute (Siehe Objektsequenz: Objekt-Attribute) Ersatzwert: A0
T *fonttyp*	Zeichenfont Ersatzwert: COURIER08f
C 1 ... 255	Y-Faktor Ersatzwert: C1
D 1 ... 255	X-Faktor Ersatzwert: D1
F 1 ... 255	Zeichenabstand in Anzahl Dots Ersatzwert: F1
P -99 .. +99	Abstand zum Barcode Bezugspunkt ist die letzte Dotlinie des Barcodesymbols. Ein negativer Wert plaziert die Subscriptzeile in das Barcodesymbol hinein, positive Werte entfernen die Subscriptzeile vom Barcode.
P %	**Keine** Subscriptzeile drucken Ersatzwert: P1



barcodedaten :

gültig: 0 - 9 (nur numerisch)

Code 2of5 Interleaved verlangt eine geradzahlige Anzahl an Ziffern.
Ist die
eingegebene Anzahl an Ziffern ungerade, so wird automatisch eine
führen de Null ergänzt.

Anzahl: beliebig, je nach Druckfläche

Symbolaufbau , Symbolbreite:

Ruhezone , Startzeichen, Nutzzeichen , [Prüfzeichen], Stopzeichen , Ruhezone

Ruhezone : Freifeld , Breite mind. 10-fache Elementbreite
Start-, Stopzeichen: werden automatisch vom Programm generiert
Prüfzeichen : kann automatisch vom Programm generiert werden, ist aber nicht fester Bestandteil des Code 2 of 5 Interleaved.
Das Prüfzeichen errechnet sich über eine Modulo-10 - Prüfsumme mit Gewicht 3.

Randzeichen, Nutzzeichen, Prüfzeichen beinhalten je nach Ratio folgende Anzahl an Elementbreiten:

Ratio	Randzeichen zusammen	Nutzzeichen , Prüfzeichen
R 2:1	8	7
R 3:1	9	9
R 5:2	17	16

Beispiel: Code 2 of 5 Interleaved

```
<STX>
<ESC>G50<ESC>I40<ESC>R0
<ESC>BC_2o5_I;H70;K0;B3;R3;Z1;TARIAL20;F2;P1;>12345678
<EOT>
```



Code 39 (Optional)

Objektsequenz:	<ESC>**BC_39;** [*parameter* **;** ...] **>** *barcodedaten* [<CR>]

Unterstützung optional

parameter :	**Barcode:**
H 1 ... 1000	Barcodehöhe in Anzahl Dots Ersatzwert: H120
B 1 ... 99	Breite eines schmalen Elementes in Anzahl Dots Ersatzwert: B3
R 2 | 3 | 5	Ratio = Verhältnis von breitem Element zu schmalem Element 2 = 2 : 1 , 3 = 3 : 1 , 5 = 5 : 2 Ersatzwert: R3
K 0 | 1	Kontrollinie: 0 = OHNE , 1 = MIT Kontrollinie Ersatzwert: K0
Z 0 | 1 | 2	Generierung bzw. Ausdruck des Prüfzeichens in der Subscriptzeile 0 = kein Prüfzeichen generieren 1 = Prüfzeichen im Barcode, aber nicht in der Subscriptzeile 2 = Prüfzeichen im Barcode und in der Subscriptzeile Ersatzwert: Z0
	Subscriptzeile:
A *d*	Objekt-Attribute (Siehe Objektsequenz: Objekt-Attribute) Ersatzwert: A0
T *fonttyp*	Zeichenfont Ersatzwert: COURIER08f
C 1 ... 255	Y-Faktor Ersatzwert: C1
D 1 ... 255	X-Faktor Ersatzwert: D1
F 1 ... 255	Zeichenabstand in Anzahl Dots Ersatzwert: F1
P -99 .. +99	Abstand zum Barcode Bezugspunkt ist die letzte Dotlinie des Barcodesymbols. Ein negativer Wert plaziert die Subscriptzeile in das Barcodesymbol hinein, positive Werte entfernen die Subscriptzeile vom Barcode.
P %	**Keine** Subscriptzeile drucken Ersatzwert: P1

barcodedaten :

gültig: 0 - 9 , A - Z
Leerzeichen - . $ / + %

Anzahl: beliebig, je nach Druckfläche

Symbolaufbau , Symbolbreite:

Ruhezone , Randzeichen, Nutzzeichen , [Prüfzeichen], Randzeichen , Ruhezone

Ruhezone : Freifeld , Breite mind. 10-fache Elementbreite
Randzeichen: werden automatisch vom Programm generiert
Prüfzeichen : kann automatisch vom Programm generiert werden, ist aber nicht fester Bestandteil des Code 39. Das Prüfzeichen errechnet sich über eine Modulo-43 - Prüfsumme.

Randzeichen, Nutzzeichen, Prüfzeichen beinhalten je nach Ratio folgende Anzahl an
Elementbreiten:
R 2:1 = 13 Elemente , R 3:1 = 16 Elemente , R 5:2 = 29 Elemente

Beispiel: Code 39

```
<STX>
<ESC>G50<ESC>I40<ESC>R0
<ESC>BC_39;H70;K0;B3;R3;Z1;TARIAL20;F2;P1;>CODE39
<EOT>
```



Code 128 (Optional)

Objektsequenz: <ESC>**BC_128;** [*parameter* **;** ...] **>** *barcodedaten* [<CR>]

Unterstützung optional

parameter :	**Barcode:**
H 1 ... 1000	Barcodehöhe in Anzahl Dots Ersatzwert: H120
B 1 ... 99	Breite eines schmalen Elementes in Anzahl Dots Ersatzwert: B3
K 0 | 1	Kontrollinie: 0 = OHNE , 1 = MIT Kontrollinie Ersatzwert: K0
Z 1 | 2	Ausdruck von Startcode und Prüfzeichen in der Subscriptzeile 1 = nicht mitausdrucken , 2 = mitausdrucken Ersatzwert: Z1
S 0 | a | b | c	Startzeichen : (Startzeichen kann auch durch die 1.Barcodate festgelegt werden) a = Startcode für Zeichensatz A, ansonsten keine weiteren Konvertierungen b = Startcode für Zeichensatz B, ansonsten keine weiteren Konvertierungen c = Startcode für Zeichensatz C Das Programm konvertiert die Ziffernpaare. Ist die Anzahl der Ziffern ungerade, so wird eine führende '0' ergänzt. 0 = automatische Komprimierung: Durch Umschalten auf den jeweils günstigsten Zeichensatz wird ein Barcodesymbol mit der kürzestmöglichen Breite erzeugt. Ersatzwert: S0
	Subscriptzeile:
A *d*	Objekt-Attribute (Siehe Objektsequenz: Objekt-Attribute) Ersatzwert: A0
T *fonttyp*	Zeichenfont Ersatzwert: COURIER08f
C 1 ... 255	Y-Faktor Ersatzwert: C1
D 1 ... 255	X-Faktor Ersatzwert: D1

F 1 ... 255	Zeichenabstand in Anzahl Dots Ersatzwert: F1
P -99 .. +99	Abstand zum Barcode Bezugspunkt ist die letzte Dotlinie des Barcodesymbols. Ein negativer Wert plaziert die Subscriptzeile in das Barcodesymbol hinein, positive Werte entfernen die Subscriptzeile vom Barcode.
P %	**Keine** Subscriptzeile drucken Ersatzwert: P1

barcodedaten :

Code 128 unterscheidet zwischen 3 verschiedenen Zeichensätzen. Die Wahl für den entsprechenden Zeichensatz kann getroffen werden mit dem 1.Barcodezeichen oder über den Parameter S*x*.
Ist das 1. Zeichen kein gültiger Startcode, so gilt die Vorgabe aus S*x* und der entsprechende Startcode wird eingefügt.

Gültige Startcodezeichen :

Startcode A : ç (135 dezimal)
Startcode B : ê (136 dezimal)
Startcode C : ë (137 dezimal)

nachfolgende Zeichen :

Zeichensatz C : 0 - 9 , jedoch gerade Anzahl Ziffern, sonst wird eine führende '0' ergänzt

Zeichensatz A , B : alle ASCII-Zeichen im Bereich 32 - 127 :
0 - 9 , A - Z , a - z
Leerzeichen ! " # $ % & ' ( ) * + , - . / : ; < = >? @
[\ ] ^ _ { | } ~ DEL

Steuerzeichen in

Zeichensatz A	Zeichensatz B	zu sendender ASCII-Code
FNC3	FNC3	Ç (128 dezimal)
FNC2	FNC2	ü (129)
SHIFT	SHIFT	é (130)
Code C	Code C	â (131)
Code B	FNC4	ä (132)
FNC4	Code A	à (133)
FNC1	FNC1	å (134)

Automatik : wie Zeichensatz B , jedoch ohne: SHIFT , Code A, Code C

Anzahl: beliebige Anzahl Zeichen, je nach Druckfläche.

Symbolaufbau , Symbolbreite:

Ruhezone , Startcode , Nutzzeichen , Prüfzeichen , Stopzeichen , Ruhezone

Ruhezone :	Freifeld , Breite mind. 10-fache Elementbreite
Startcode :	s.o.
Stopcode :	wird automatisch vom Programm generiert
Prüfzeichen :	wird automatisch vom Programm generiert

Startcode, Nutzzeichen, Prüfzeichen bestehen jeweils aus 11 Elementbreiten.
Das Stopzeichen hat die 13-fache Breite des schmalen Elements.

Beispiel: Code 128

```
<STX>
<ESC>G50<ESC>I40<ESC>R0
<ESC>BC_128;H70;K0;B3;TARIAL20;F2;P1;>Code128
<EOT>
```



EAN-8 (Optional)

Objektsequenz: <ESC>**BEAN8;** [*parameter* **;** ...] **>** *barcodedaten* [<CR>]

Unterstützung optional

parameter :	**Barcode:**
H 1 ... 1000	Barcodehöhe in Anzahl Dots Ersatzwert: H120
B 1 ... 4	Breite eines schmalen Elementes in Anzahl Dots Ersatzwert: B3 (Zusätzlich kann das gesamte Barcodesymbol über die Objektsequenzen: <ESC> C ... , <ESC> D ... in Höhe und Breite vervielfacht werden.)
K 0 | 1	Kontrollinie: 0 = OHNE , 1 = MIT Kontrollinie Ersatzwert: K0

	Subscriptzeile:
P 1	Subscriptzeile mitausdrucken
P %	**Keine** Subscriptzeile drucken
	Ersatzwert: P1

barcodedaten :

gültig: 0 - 9 (nur numerisch)

Anzahl: Der Barcode besteht aus 8 Ziffern einschließlich Prüfziffer.
Bei Eingabe von 8 Ziffern wird die 8. Ziffer als Prüfziffer kontrolliert.
Bei Eingabe von 7 Ziffern wird die Prüfziffer berechnet und ergänzt.

Symbolaufbau , Symbolbreite:

Ruhezone , Randzeichen, 4 Nutzziffern , Trennzeichen, 4 Nutzziffern, Randzeichen , Ruhezone

Ruhezone : Freifeld , Breite mind. 10-fache Elementbreite
Randzeichen:: werden automatisch vom Programm generiert

Ein EAN-8 - Barcodesymbol besteht aus insgesamt 67 Elementbreiten.

Beispiel: EAN-8

```
<STX>
<ESC>G50<ESC>I40<ESC>R0
<ESC>BEAN8;H70;K0;B3;>4012345
<EOT>
```



EAN-13 (Optional)

Objektsequenz: <ESC>**BEAN13;** [*parameter* **;** ...] **>** *barcodedaten* [<CR>]

Unterstützung optional

parameter :

	Barcode:
H 1 ... 1000	Barcodehöhe in Anzahl Dots Ersatzwert: H120
B 1 ... 4	Breite eines schmalen Elementes in Anzahl Dots Ersatzwert: B3 (Zusätzlich kann das gesamte Barcodesymbol über die Objektsequenzen: <ESC> C ... , <ESC> D ... in Höhe und Breite vervielfacht werden.)
K 0 | 1	Kontrollinie: 0 = OHNE , 1 = MIT Kontrollinie Ersatzwert: K0
	Subscriptzeile:
P 1	Subscriptzeile mitausdrucken
P %	**Keine** Subscriptzeile drucken Ersatzwert: P1

barcodedaten :

gültig: 0 - 9 (nur numerisch)
Ein Blank vor der 1.Ziffer bewirkt, dass die 1. Ziffer der Subscriptzeile links neben dem Barcode gedruckt wird.
Ohne Blank werden alle 13 Ziffern unterhalb des Barcodes gedruckt.

Anzahl: Der Barcode besteht aus 13 Ziffern einschließlich Prüfziffer.
Bei Eingabe von 13 Ziffern wird die 13. Ziffer als Prüfziffer kontrolliert.
Bei Eingabe von 12 Ziffern wird die Prüfziffer berechnet und ergänzt.





Symbolaufbau , Symbolbreite:

Ruhezone , Randzeichen, 7 Nutzziffern , Trennzeichen, 6 Nutzziffern, Randzeichen , Ruhezone

Ruhezone : Freifeld , Breite mind. 10-fache Elementbreite
Randzeichen:: werden automatisch vom Programm generiert

Ein EAN-13 - Barcodesymbol besteht aus insgesamt 95 Elementbreiten bzw. aus 106 Elementbreiten, wenn die 1. Ziffer links neben dem Barcode gedruckt wird.

Beispiel: EAN-13

```
<STX>
<ESC>G50<ESC>I40<ESC>R0
<ESC>BEAN13;H70;K0;B3;> 401234567890
<EOT>
```



EAN-128 (Optional)

Objektsequenz: <ESC>**BEAN128 ;** [*parameter* **;** ...] **>** *barcodedaten* [<CR>]

Unterstützung optional

parameter : **Barcode:**

H 1 ... 1000 — Barcodehöhe in Anzahl Dots
Ersatzwert: H120

B 1 ... 99 — Breite eines schmalen Elementes in Anzahl Dots
Ersatzwert: B3

K 0 | 1 — Kontrollinie: 0 = OHNE , 1 = MIT Kontrollinie
Ersatzwert: K0

Z 1 | 2 — Ausdruck von Startcode und Prüfzeichen in der Subscriptzeile
1 = nicht mitausdrucken , 2 = mitausdrucken
Ersatzwert: Z1

S 0 | a | b | c — Startzeichen :
(Startzeichen kann auch durch die 1.Barcodate festgelegt werden)
a = Startcode für Zeichensatz A,
ansonsten keine weiteren Konvertierungen
b = Startcode für Zeichensatz B,
ansonsten keine weiteren Konvertierungen
c = Startcode für Zeichensatz C
Das Programm konvertiert die Ziffernpaare.
Ist die Anzahl der Ziffern ungerade, so wird eine führende '0' ergänzt.
0 = automatische Komprimierung:
Durch Umschalten auf den jeweils günstigsten Zeichensatz wird ein Barcodesymbol mit der kürzestmöglichen Breite erzeugt. Statt S0 kann dieser Parameter auch entfallen.
Ersatzwert: S0

Subscriptzeile:

A *d* — Objekt-Attribute (Siehe Objektsequenz: Objekt-Attribute)
Ersatzwert: A0
T *fonttyp* Zeichenfont
Ersatzwert: COURIER08f

C 1 ... 255 — Y-Faktor
Ersatzwert: C1

D 1 ... 255	X-Faktor Ersatzwert: D1
F 1 ... 255	Zeichenabstand in Anzahl Dots Ersatzwert: F1
P -99 .. +99	Abstand zum Barcode Bezugspunkt ist die letzte Dotlinie des Barcodesymbols. Ein negativer Wert plaziert die Subscriptzeile in das Barcodesymbol hinein, positive Werte entfernen die Subscriptzeile vom Barcode.
P %	**Keine** Subscriptzeile drucken Ersatzwert: P1

barcodedaten :

Der EAN-128 ist eine Variante des Code 128. Beim EAN-128 folgt nach dem Startcode immer das Codezeichen: FNC1. Dieses Codezeichen wird automatisch vom Druckerprogramm generiert.
FNC1 kann aber auch als Abschlußzeichen für variable Felder innerhalb des Barcodestrings benutzt werden. In diesem Fall muss es als Date miteingegeben werden.

Je nach Zeichensatz (Sx) sind folgende Zeichen gültig :

Zeichensatz C : (Sc) 0 - 9

Zeichensatz A/B (Sa/Sb): alle ASCII-Zeichen im Bereich 32 - 127 :
0 - 9 , A - Z , a - z
Leerzeichen ! " # $ % & ' ( ) * + , - . / : ; < = >?
@ [\ ] ^ _ { | } ~ DEL

Steuerzeichen in

Zeichensatz A	Zeichensatz B	zu sendender ASCII-Code
FNC3	FNC3	Ç (128 dezimal)
FNC2	FNC2	ü (129)
SHIFT	SHIFT	é (130)
Code C	Code C	â (131)
Code B	FNC4	ä (132)
FNC4	Code A	à (133)
FNC1	FNC1	å (134)

autom. Komprimierung (S0) : ASCII-Zeichen 32-127 + FNC1

Anzahl: max. 48 Nutzzeichen, jedoch darf die Anzahl der Codezeichen (Start-, Stop-, Steuer-, Prüf- und codierte Nutzzeichen) nicht größer als 35 sein..

Symbolaufbau , Symbolbreite:

Ruhezone , Startcode , FNC1 , Nutzzeichen , Prüfzeichen , Stopzeichen ,



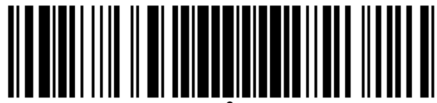
Ruhezone

Ruhezone : Freifeld , Breite mind. 10-fache Elementbreite
Startcode : s.o.
FNC1: dieses Codezeichen wird automatisch eingefügt
Stopcode : wird automatisch vom Programm generiert
Prüfzeichen : wird automatisch vom Programm generiert (Modulo 103)

Startcode, Nutzzeichen, Prüfzeichen bestehen jeweils aus 11 Elementbreiten.
Das Stopzeichen hat die 13-fache Breite des schmalen Elements.

Beispiel: EAN 128

```
<STX>
<ESC>G50<ESC>I40<ESC>R0
<ESC>BEAN128;H70;K0;B3;TARIAL20;F2;P1;>106593å211678
<EOT>
```



PDF-417 (Optional)

Objektsequenz: <ESC>**BPDF417;** [*parameter* **;** ...] D *barcodedaten* [<CR>]

Unterstützung optional

parameter **Bedeutung**:

L **Angabe des ERROR-Levels**.

Ein PDF417 beinhaltet einen Fehlererkennungscode, also eine Art Prüfziffer. Zusätzlich kann noch ein Fehlerkorrekturcode angehängt werden,sodass unleserliche Codewörter wiederhergestellt werden können. Je höher der ERROR-Level, desto höher der Wiederherstellungsgrad.

Der Grad des ERROR-Levels sollte jedoch maßvoll gewähltwerden, da ein hoher Grad nicht nur das Barcodesymbol vergrößert, sondern auch die Anzahl der Nutzdaten mindert.
Empfehlenswert: 10% der Anzahl der Codewörter für die Nutzdaten sollten
für die Fehlererkennung geopfert werden.
Folgende ERROR-Levels sind möglich:

L0	:	nur Fehlererkennung, keine Fehlerkorrektur		
L1	:	Fehlererkennung	+ 2	Codewörter für Korrektur
L2	:	"	+ 6	"
L3	:	"	+ 14	"
L4	:	"	+ 30	"
L5	:	"	+ 62	"
L6	:	"	+ 126	"
L7	:	"	+ 254	"
L8	:	"	+ 510	"

(Ein PDF-Symbol kann aus max. 925 Codewörter bestehen.)

Die Angabe des ERROR-Levels kann auch in % erfolgen:

L%d : d gibt den prozentualen Anteil der Korrekturcode-
wörter zu den Nutzdatencodewörtern an.
(Bsp.: L%10)

Wird die L-Angabe nicht gemacht, so wird L%10 angenommen. Sollte eine feste Matrix für das Barcodesymbol vorgegeben werden, so wird der Korrekturgrad soweit erhöht, wie freie Codewörter übrig sind.

C **Anzahl der Codespalten**

Definiert die Breite des PDF-Symbols.
Ein PDF-Symbol besteht aus Randzeichen, Zeilenindikatoren und den

Codewörtern für die Nutzdaten (=Codespalten).
Jedes Codewort besteht aus 17 Modulen, wobei ein Modul mindestens 1/6 mm breit sein sollte. Randzeichen und Zeilenindikatoren zusammen sind 69 Module breit.

Mögliche Angaben für PDF-Codespalten: **C1 ... C30**.

Dieser Parameter ist wahlfrei. Wird er nicht angegeben, so muss der R-Parameter definiert werden !
Das Programm ermittelt die erforderliche Breite in Abhängigkeit von der Anzahl
der Nutzdaten und des gewählten Korrekturgrades.

R **Anzahl der Codezeilen**
Definiert die Höhe des PDF-Symbols.
Ein PDF-Symbol besteht aus mehreren übereinander angeordneten Barcodezeilen : 3 ... 90.

Mögliche Angaben für PDF-Codezeilen: **R3 ... R90**.

Dieser Parameter ist wahlfrei. Wird er nicht angegeben, so muss der C-Parameter definiert werden !
Das Programm ermittelt die erforderliche Höhe in Abhängigkeit von der Anzahl
der Nutzdaten und des gewählten Korrekturgrades.

R , C **Definition einer Symbolmatrix**
Werden C- und R-Parameter definiert, so ist die Symbolgröße immer entsprechend der Definition gleich groß.
Werden weniger Nutzdaten angegeben als im Symbol codiert werden könnten, so wird mit Füllzeichen aufgefüllt.
Zuviele Nutzdaten würden allerdings einen Abbruch mit Fehler provozieren.

T **Truncated PDF**
Darstellung des PDF-Symbols in einer verkürzten Form.
Hierbei wird der rechte Zeilenindikator und das rechte Randzeichen durch ein Modul dargestellt. Die Anzahl der Module für Zeilenindikatoren und Randzeichen beträgt hier nur noch 35 Module. Diese Kurzform sollte nicht benutzt werden, wenn eine hohe Fehlerkorrektur erforderlich ist.
T0 = normaler Modus (= Voreinstellung)
T1 = Truncated PDF

W **Breite eines Moduls**
Definiert die Modulbreite in Anzahl Dots.
Ein Modul sollte mindestens 1/6 mm breit sein (2 Dots bei einer 12-Dot-Thermoplatine).
Bsp.: W2 = 2 Dots / Modul
Voreingestellter Wert ist: W2.

H **Höhe eines Moduls (Barcodezeile)**
Definiert die Höhe einer Barcodezeile in Anzahl Dots.
Die Höhe einer Barcodezeile sollte mindestens das 3-4 fache der Modul-breite betragen.

Bsp.: H6 = 6 Dots / Barcodezeile
Voreingestellter Wert ist: H6.

D **PDF - Nutzdaten**

In einem PDF-Symbol können je nach Datenart unterschiedlich viele Daten codiert werden. Hierbei werden die Daten in sogenannte Codewörter umgewandelt.
Ein PDF-Symbol kann maximal 925 Codewörter beinhalten.
Von diesen 925 Codewörtern muss noch die Anzahl Codewörter für die Fehlerkorrektur abgezogen werden.
In einem Codewort kann codiert werden:

2 alphanumerische Daten
2,93 numerische Daten
1,2 Binärdaten (Wertebereich: 0 - 255)

Bei einem eingestellten ERROR-Level 0 könnten also maximal in einem PDF-Symbol codiert werden:

1850 alphanumerische Daten
2710 numerische Daten
1108 Binärdaten

Die Druckersoftware sorgt für die richtige Codierung der einzelnen Datenarten. Es erfogt eine Optimierung derart, dass die Codewortanzahl so klein wie möglich gehalten wird.

Eingabe der Nutzdaten:

Gültig sind alle Daten im Wertebereich: 0 - 255.
Um Konflikte mit anderen Druckersteuerzeichen auszuschließen und zur Endeerkennung der Nutzdaten müssen alle Daten, deren ASCII-Wert kleiner als 32 ist, in einer speziellen Form eingegeben werden:

\ddd (ddd = 3-stelliger Dezimalwert)
\\ (für Zeichen: "\")
Bsp.: \013 (= "CR")

PDF417 als variables Datenfeld.

Der PDF417 kann, wie andere Barcodes auch, als variables Barcodefeld verwendet werden.

Fehlermeldungen:

Treten Konflikte auf bei der Darstellung des PDF-Symbols infolge von fehlerhaften Parametern, so wird folgende Fehlermeldung erzeugt:

FEHLER #074
PDF 417 - Daten

Mögliche Fehlerursachen :

- falscher ERROR-Level
- falsche Angabe für Spaltenanzahl, Barcodezeilenanzahl
- Anzahl Nutzdaten zu groß
- kein D-Parameter bzw. keine Nutzdaten
- weder C-Parameter noch R-Parameter definiert
- Anzahl Daten-Codewörter + Anzahl Korrektur-Codewörter größer 925

Nach dieser Fehlermeldung ist ein Ausdruck nicht mehr möglich.

Beispiel:

```
<SX>
<EC>G020<EC>I0020<EC>V1
<EC>OPDF417;L%10;C3;T0;W2;H8;
D Dies ist ein PDF417-Barcode.\013\010
PRINTON GmbH\013\010<CR>
<ET>
```

Hierbei handelt es sich um einen PDF417-Barcode, der zusätzlich als variables Barcodefeld (#1) definiert wurde.
Die Barcodebreite ist festgelegt auf 3 Codespalten. (= 3*17 + 69 = 120 Module) Jedes Modul ist 2 Dots breit --> Barcodebreite = 120 * 2 = 240 Dots.

Das Programm berechnet die einzelnen Codewörter und fügt noch 10% der Codewortanzahl für die Korrektur hinzu. So ergibt sich die Anzahl der Barcodezeilen.
Die Höhe jeder Barcodezeile ist festgelegt auf 8 Dots.

Variables Barcodefeld:

Um ein weiteres Etikett mit neuen PDF-Daten zu drucken, kann beispielsweise folgender variabler Steuersatz zum Drucker gesendet werden:

<ESC>v1; Neue PDF417-Daten.\013\010 <CR>
<ESC>#1<CR>

Nur die PDF-Daten sind geändert. Das Kartenlayout bleibt erhalten.

Bei Eingabe des variablen Steuersatzes : <ESC>v1;<CR>
würde kein PDF-Symbol gedruckt werden.

DATAMATRIX (Optional)

Objektsequenz: <ESC>**BDataMatrix ;** [*parameter* **;** ...] **>** *barcodedaten* [<CR>]

Unterstützung optional

parameter :

M 0..24

Barcode:

Datamatrix Size.

Angabe des Aufbaus der Datamatrix. Je nach Aufbau der Datamatrix können 3 bis zu 1558 Datenworte gespeichert werden.

Ersatzwert: M0

Definition der Parameter:

M0 Automatische Ermittlung des Matrixgröße, je nach Anzahl der Barcodedaten.

Parameter	Size	Daten	Error
M1	10x10	3	5
M2	12x12	5	7
M3	14x14	8	10
M4	16x16	12	12
M5	18x18	18	14
M6	20x20	22	18
M7	22x22	30	22
M8	24x24	36	24
M9	26x26	44	28
M10	32x32	62	36
M11	36x36	86	42
M12	40x40	114	48
M13	44x44	144	56
M14	48x48	174	68
M15	52x52	204	84
M16	64x64	280	112
M17	72x72	368	144
M18	80x80	456	192
M19	88x88	576	224
M20	96x96	696	272
M21	104x104	816	336
M22	120x120	1050	408
M23	132x132	1304	496
M24	144x144	1558	620

T 0..1 Bevorzugte Codierungsart.

Je nach Codierungsart, ist es möglich das die Anzahl der Datenbytes sich von den unterschiedlichen Codierungsarten unterscheidet.

Die default Codierungsart ist die ASCII Codierung. Hierbei wird mindestens ein Datenwort für jedes Zeichen aus den übergebenen Barcodedaten benötigt. Hinzu kommen eventuell noch Steuerzeichen. Bevorzugt kann die C40 Codierungsart verwendet werden. Bei dieser Codierungsart, werden bis zu drei Barcodedaten in zwei Datenworte abgelegt.

Ersatzwert: T0

Definition der Parameter:

T0 : Bevorzugt ASCII Codierung

T1 : Bevorzugt C40 Codierung (Zur Zeit nicht unterstützt)

C 1 ... 255 Y-Faktor

Ersatzwert: C1

D 1 ... 255 X-Faktor

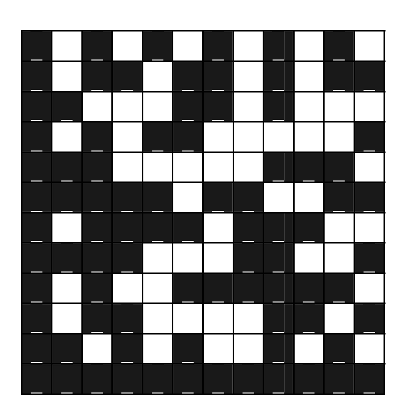
Ersatzwert: D1

barcodedaten :

Der Datamatrix ist ein 2 Dimensionaler Barcode. Innerhalb des Datamatrixcodes werden gleichzeitig Korrekturwerte mit abgelegt. Wir benutzen für die Ermittlung der Korrekturwerte eine Reed-Solomon Fehler-Korrektur (ECC200)

Beispiel: DATAMATRIX

```
<STX>
<ESC>G50<ESC>I40<ESC>R0
<ESC>BDataMatrix;M0;T0;C1;D1>Hallo
<EOT>
```



i. Die serielle Schnittstelle

Einstelloptionen der seriellen Schnittstelle

Der Drucker verfügt standardmäßig über eine serielle Schnittstelle für die Kommunikation mit Computer, die einen seriellen Anschluß erfordern. Sie unterstützt 2 Arten von Handshake-Protokollen und kann mit 2 Übertragungsgeschwindigkeiten arbeiten.

Werksseitig ist die Schnittstelle wie folgt eingestellt:

19200 Baud
8 Datenbits
1 Stopbit
ohne Parität

Die Baudrate kann über das Menü auf 9600 oder 38400 Baud geändert werden. Weitere Einstellmöglichkeiten sind über die Steuersequenz <ESC>f... möglich.

Übertragungsprotokolle der seriellen Schnittstelle

Die serielle Schnittstelle bietet zwei Datenübertragungsprotokolle: Betriebsbereitschaft (DATA TERMINAL READY) und XON/XOFF. Bei dem ersten handelt es sich um ein Hardware-, bei dem zweiten um ein Software-Protokoll. Über diese Übertragungsprotokolle wird sichergestellt, dass der Rechner nicht schneller Daten an den Drucker sendet, als dieser aufnehmen kann.

Betriebsbereitschaft (DTR)

Mit diesem Übertragungsprotokoll werden die Signale an den Pins der Schnittstelle gesteuert. Die Anschlußbelegung ist in der nachfolgenden Tabelle dargestellt. Das DTR-Pin steuert den Informationsfluß zum Drucker.
Sobald die verbleibende Pufferkapazität unter 256 Bytes fällt, ändert der Drucker das DTR-Signal auf LOW-Pegel.
Sind mehr als 512 Bytes frei, so wechselt das Signal wieder zu einem HIGH-Pegel über.

Innerhalb von 256 Zeichen nach Ändern des DTR-Signals auf LOW durch den Drucker, muss der Computer die Datenübertragung stoppen, andernfalls kommt es zum Speicherüberlauf und die Daten gehen verloren (→ FEHLER: Eingabespeicher).

XON / XOFF

Hierbei handelt es sich um das einfachste Softwareprotokoll. Sobald die verbleibende Pufferkapazität unter 256 Bytes fällt, wird der ASCII-Code XOFF (13 hex) zum Computer übertragen, wobei dieser aufgefordert wird, die Datenübertragung zu stoppen.
Sobald sich im Eingabespeicher wieder mehr als 512 freie Bytes befinden, sendet der Drucker den ASCII-Code XON (12 hex) und weist den Computer dadurch an, die Übertragung fortzusetzen.
Falls der Computer innerhalb 256 Zeichen nach Empfang eines XOFF-Codes nicht mit der Datenübertragung stoppt, kommt es zum Speicherüberlauf und die Daten gehen verloren.

Das XON-/XOFF-Protokoll wird über <ESC>f... aktiviert.

Anschlußbelegung der seriellen Schnittstelle

Schnittstellenanschlüsse Druckerseite RJ45

Pin	Signal	Sender	Funktion
1,2	5V		5 Volt Versorgungsspannung
3	CTS	Computer	Computer bereit für Datenempfang
4	RTS	Printer	Übertragungsanforderung, immer HIGH
5	RxD	Computer	Daten empfangen. Empfängt die vom Computer übertragenen Daten.
6	TxD	Printer	Daten senden. Sendet die Daten an den Computer.
7,8	GND		Signalerde

Signalpegel der seriellen Schnittstelle:

Signalpegel	Eingang	Ausgang
LOW-Pegel	-25V bis - 3V	- 12V
HIGH-Pegel	+3V bis + 25V	+ 12V

ii. Warnungen, Hinweise, Fehler

Fehlerebene 1 - Warnung

Meldung	Mögliche Ursachen	Maßnahmen
WARNUNG #001	Falscher Wert nach <ESC># Es darf nur ein numerischer Wert eingegeben werden.	Sequenz korrigieren
WARNUNG #002 Steuersequenz	Falscher Wert nach <ESC>b: ■ unzulässiges Zeichen ■ Wert < 150 oder ■ Wert > 2520	Sequenz korrigieren
WARNUNG #003 Steuersequenz	Falscher Wert nach <ESC>c: ■ unzulässiges Zeichen ■ Wert < 64 und Wert > Dotanzahl Thermokopf	Sequenz korrigieren
WARNUNG #004	Falscher Wert nach <ESC>d: ■ Wert > 240	Sequenz korrigieren
WARNUNG #005	Falscher Wert nach <ESC>e: ■ Wert < 320	Sequenz korrigieren
WARNUNG #006 bis #009	***Wird zur Zeit nicht verwendet !***	
WARNUNG #010 Steuersequenz	Falscher Wert nach <ESC>j: ■ unzulässiges Zeichen	Sequenz korrigieren
WARNUNG #011 Steuersequenz	Falscher Wert nach <ESC>k: ■ Schalter ungleich 0 oder 1 ■ Label-Taken nicht erlaubt, da im Setup nicht angemeldet ■ Label-Taken und Abschneider wurden gleichzeitig angemeldet ■ Abschneiden nicht erlaubt, da Abschneider nicht angemeldet ■ Transfer-Druck nicht erlaubt, da im Setup nicht angemeldet ■ Etikettenzähler nicht erlaubt. ■ Flag Papier mit bedruckter Rückseite nicht erlaubt.	→ Sequenz korrigieren → Service → Label-Taken oder Abschneider abmelden → Abschneider anmelden → Service
WARNUNG #013	Falscher Wert nach <ESC>m	Sequenz korrigieren
WARNUNG #014 Steuersequenz	Falscher Wert nach <ESC>n: ■ Falscher Wert	Sequenz korrigieren

Meldung	Mögliche Ursachen	Maßnahmen
WARNUNG #015	***Wird zur Zeit nicht verwendet***	

WARNUNG #019	Falscher Wert oder Fehler nach <ESC>s	Sequenz korrigieren
WARNUNG #020	Falscher Wert nach <ESC>t	Sequenz korrigieren
WARNUNG #021	Falscher Wert nach <ESC>u	Sequenz korrigieren
WARNUNG #022	Falscher Wert nach <ESC>v	Eventuell passt das variable Datenelement nicht in den dafür vorgesehenen Speicherbereich
WARNUNG #023	***Wird zur Zeit nicht verwendet***	
WARNUNG #024	Falscher Wert nach <ESC>x	Der angegebene Wert ist zu gross bzw. zu klein. Wert prüfen.
WARNUNG #025	***Wird zur Zeit nicht verwendet***	
WARNUNG #026	Falscher Wert nach <ESC>z	Sequenz korrigieren
WARNUNG #027 Steuersequenz	Unzulässige Steuersequenz: Falscher Kleinbuchstabe nach <ESC>	Sequenz korrigieren
WARNUNG #028 Objekt-Kennung ?	Objekt-Kennung wurde nicht gefunden	Objekt mit einer Kennung über <ESC>V versehen. Objekt mit gleicher Kennung über <ESC>v neue Daten zuweisen.
WARNUNG #029 var. Logolänge	Anzahl var. Logodaten nicht identisch mit Orginal-Logo	Var. Logo wird ignoriert. Var. Logo auf gleiches Format wie Orginal-Logo bringen.
WARNUNG #030 bis #032	***Wird zur Zeit nicht verwendet !***	
WARNUNG #033 Objektsequenz	Falscher Wert nach <ESC>C: ■ Y-Faktor = 0 ■ Y-Faktor > 255	Sequenz korrigieren. Ersatzwert: einfache Zeichenhöhe
WARNUNG #034 Objektsequenz	Falscher Wert nach <ESC>D: ■ X-Faktor = 0 ■ X-Faktor > 255	Sequenz korrigieren. Ersatzwert: einfache Zeichenbreite
WARNUNG #035	***Wird zur Zeit nicht verwendet***	
WARNUNG #036 Objektsequenz	Falscher Wert nach <ESC>F: ■ Zeichenabstand > 255	Sequenz korrigieren. Ersatwert: Zeichenabstand = 1
WARNUNG #037 Objektsequenz	Falscher Wert nach <ESC>G: ■ X-Position = 0 ■ X-Position außerhalb des Bildbereichs	Sequenz korrigieren. Die X-Position darf nicht größer als die Bildbreite sein. Ersatzwert: X-Position = 1

Meldung	Mögliche Ursachen	Maßnahmen
WARNUNG #038	***Wird zur Zeit nicht verwendet***	
WARNUNG #039 Objektsequenz	Falscher Wert nach <ESC>I: ■ Y-Position = 0 ■ Y-Position außerhalb des Bildbereichs	Sequenz korrigieren. Die Y-Position darf nicht größer als die Bildhöhe sein. Ersatzwert: Y-Position = 1
WARNUNG #040 bis #042	***Wird zur Zeit nicht verwendet !***	
WARNUNG #043	Flascher Wer nach <ESC>M: ■ Falscher Logoname	Sequenz korrigieren oder Logo nicht im Flash vorhanden. Sequenz wird ignoriert.
WARNUNG #044 bis #046	***Wird zur Zeit nicht verwendet !***	
WARNUNG #047 Objektsequenz	Falsche Angabe(n) nach <ESC>Q: ■ Fortschaltungswert oder Fort-schaltungszyklus fehlen ■ Fortschaltungswert > +/- 9 ■ Fortschaltungszyklus = 0 oder >255 ■ Schalter für Nullunterdrückung ungleich 0 oder 1 ■ Beginn des Fortschaltungfeldes < 1 ■ Größe des Fortschaltungfeldes < 0 ■ Zweiter Parameter fehlt.	Sequenz korrigieren
WARNUNG #048 bis #051	***Wird zur Zeit nicht verwendet !***	
WARNUNG #052 Objektsequenz	Falsche Angabe nach <ESC>V: ■ Die Kennung besteht aus mehr als 1 Zeichen	Sequenz korrigieren. Erlaubt ist nur 1 Zeichen als Kennung. Ersatzwert: Das Objekt erhält keine Kennung.
WARNUNG #053	***Wird zur Zeit nicht verwendet***	
WARNUNG #054 Objektsequenz	Falscher Wert nach <ESC>X: ■ Koordinaten falsch	Sequenz korrigieren.

Meldung	Mögliche Ursachen	Maßnahmen
WARNUNG #055 Bildzeile zuviel	Bildzeilen-Überlauf. Es wurden zuviel <ESC>Y-Sequenzen an den Drucker gesendet.	Anzahl Bild-Sequenzen überprüfen. Sie darf nicht größer als die Anzahl Bildzeilen im Bildbereich sein. Ersatzwert: Überflüssige Bildzeilen werden ignoriert.
WARNUNG #056 Bildzeilenlänge	Innerhalb <ESC>Y wurden zuviel oder zuwenig Bilddaten übertragen. Es fehlt ein <CR>	Die Anzahl Bilddaten muss mit der Bildbreite / 8 übereinstimmen.
WARNUNG #057 Objektsequenz	Unzulässige Objektsequenz: Falscher Großbuchstabe nach<ESC>	Sequenz korrigieren
WARNUNG #058 Objekt > 64KByte	Objekt benötigt mehr als 64KByte Speicherplatz	Objekt verkleinern oder in 2 Objekte aufteilen. ***Wird innerhalb des PE120 zur Zeit nicht verwendet !***
WARNUNG #059	***Wird zur Zeit nicht verwendet***	
WARNUNG #060 Font-Typ	Fehlerhafte <ESC>T-Sequenz: Angewählter Font-Typ ist nicht im Drucker vorhanden.	Vorhandener Font-Typ anwählen. Ersatztyp: Courier 08 fett
WARNUNG #061 Barcode-Typ	Fehlerhafte <ESC>B-Sequenz: Angewähler Barcode-Typ ist nicht im Drucker vorhanden.	Anderer Barcode-Typ wählen oder Sequenz entfernen. Ersatztyp: keinen, Sequenz wird ignoriert.
WARNUNG #062 Code 2of5 – Daten	Fehlerhafte <ESC>B-Sequenz: Unzulässige Barcodedaten	Sequenz korrigieren
WARNUNG #063 Code 39 – Daten	Fehlerhafte <ESC>B-Sequenz: Unzulässige Barcodedaten	Sequenz korrigieren
WARNUNG #064 Code 128 – Daten	Fehlerhafte <ESC>B-Sequenz: Unzulässige Barcodedaten	Sequenz korrigieren
WARNUNG #065 EAN 8 – Daten	Fehlerhafte <ESC>B-Sequenz: Unzulässige Barcodedaten	Sequenz korrigieren
WARNUNG #066 EAN 13 – Daten	Fehlerhafte <ESC>B-Sequenz: Unzulässige Barcodedaten	Sequenz korrigieren
WARNUNG #067 UPC-A - Daten	Fehlerhafte <ESC>B-Sequenz: Unzulässige Barcodedaten	Sequenz korrigieren

Meldung	Mögliche Ursachen	Maßnahmen
WARNUNG #068 PDF417 – Daten	Fehlerhafte <ESC>B-Sequenz: Unzulässige Barcodedaten	Sequenz korrigieren
WARNUNG #069 Barcode – Daten	Fehlerhafte <ESC>B-Sequenz: Unzulässige Barcodedaten	Sequenz korrigieren
WARNUNG #070 Zeichen ?	Zeichen kann außerhalb einer ESC-Sequenz nicht zugeordnet werden.	Zeichen wird ignoriert. Auf weitere Fehlermeldungen überprüfen und diese zuerst beheben.
WARNUNG #071	Etikettentimeout Nach <STX> wurde innerhalb der definierten Zeit keine <EOT> empfangen.	Verbindung zum Drucker prüfen. Etikettenvorlage überprüfen.
WARNUNG #072	Lesefhler Barcodemodul.	Verbindung zum Drucker prüfen. Etikettenvorlage überprüfen.
WARNUNG #073 bis #079	***Wird zur Zeit nicht verwendet !***	
WARNUNG #080 Objekt -> Bild	Objekt passt von seinen Abmessungen her nicht in den Bildbereich	Objekt wird ignoriert. Objekt in seiner X-/Y-Position neu platzieren
WARNUNG #081 Drehspeicher	Nicht genügend Speicher vorhanden, um das Objekt drehen zu können.	Das Objekt wird in seiner ursprünglichen Form in das Bild kopiert. Abhilfe: Bildspeicher oder Eingabe-speicher verkleinern. Ansonsten Speicher erweitern (→ Service)
WARNUNG #082	***Wird zur Zeit nicht verwendet***	
WARNUNG #083 var. Bildspeicher	Nicht genügend Speicher für var. Bildspeicher vorhanden.	Alle var. Objekte werden ignoriert. Abhilfe: Bildspeicher oder Eingabespeicher verkleinern. Auf var. Objekte verzichten. Ansonsten Speicher erweitern (→ Service)

Meldung	Mögliche Ursachen	Maßnahmen
WARNUNG #084 Bildspeicher	Nicht genügend Speicher für Bildbereich vorhanden.	Der Bildbereich wird in seiner Höhe entsprechend dem Speicher reduziert. Dadurch werden vermutlich nicht alle Objekte dargestellt werden. Abhilfe: Bildbereich in seinen Abmessungen und Position manuell optimieren, Eingabespeicher verkleinern oder evtl. auf var. Objekte verichten. Ansonsten Speicher erweitern (→ Service)
WARNUNG #085	Modulverwaltung Warnung, kein Modul, falscher Name	Prüfen ob genügend Speicher vorhanden ist, ob Name des Moduls stimmt.
WARNUNG #086		Label Taken
WARNUNG #093 var. Papier	Papiervorende Warnung.	Diese Meldung wird gesetzt, wenn der Papierverbrauch den Level für Papiervorende Warnung überschritten hat. In der Regel wird diese Meldung ab dem Verbrauch größer gleich 80% abgesetzt. Diese Meldung wird auch dann abgesetzt, wenn ein Papiervorendeloch definiert ist. Bzw. wenn der Etikettenzähler überschritten wurde. Abhilfe: Neue Papierrolle einlegen
WARNUNG #094 var. Paper	Papierrolle hat kein Papier mehr. Bzw. Papierrolle bewegt sich nicht mehr.	Wenn diese Warnung abgesetzt wird, dann ist das Papier der Papierrolle erschöpft. Es ist nur noch möglich einen Fahrausweis zu drucken. Abhilfe: Neue Papierrolle einlegen
WARNUNG #095 bis #097	Wird zur Zeit nicht verwendet !	
WARNUNG #098	Wechselfehler. Bei Schachtwechsel ist es zu einem Problem gekommen.	Diese Warnung zeigt an, das es zu einem Problem gekommen ist. Jedoch der Schachtwechsel funktioniert hat.

Meldung	Mögliche Ursachen	Maßnahmen
WARNUNG #099	Die maximal zulässige Zeit für die Druckdatenübertragung und Ausgabe des Dokuments ist abgelaufen.	Der Drucker führt ein Internes Buffer-Reset durch. Dadurch werde alle anstehenden Daten verworfen.

Fehlerebene 2 – Fehler (Software)

Meldung	Mögliche Ursachen	Maßnahmen
WARNUNG #100 bis #109	***Wird zur Zeit nicht verwendet !***	
FEHLER #110 Objektanzahl	Es sind max. 32 variable Objekte erlaubt.	Variable Objekte auf max. 32 reduzieren. Es ist keine weitere Druckauftrags-Bearbeitung möglich. Drucker muss neu initialisieren.
FEHLER #111 Logosatz...<CR>	Endekriterium der Logo <ESC>L-Sequenz nicht erkannt.	Am Ende der Logosequenz muss ein <CR> stehen. Logodaten bzw. Logoformat überprüfen. Drucker neu initialisieren.
WARNUNG #112 bis #114	***Wird zur Zeit nicht verwendet !***	
FEHLER #115 Eingabespeicher	Eingabespeicher ist übergelaufen.	Falls kein Schnittstellen-Handling möglich ist, Eingabespeicher vergrößern, falls möglich. Drucker neu initialisieren.
WARNUNG #116 bis #149	***Wird zur Zeit nicht verwendet !***	

Fehlerebene 3 - Hinweise

Meldung	Mögliche Ursachen	Maßnahmen
WARNUNG #150 bis #159	***Wird zur Zeit nicht verwendet !***	
HINWEISE #160 Papier entnommen	Papier wurde aus dem Schacht entnommen worden.	Hinweis nur mit neu einlegen von Papier (<ESC>s[]-/+<CR> rücksetzbar.
HINWEISE #161 Papier entnommen	Papier wurde nach Entnahme Kommando nicht entnommen.	Hinweis nur mit neu einlegen von Papier (<ESC>s[]-/+<CR> rücksetzbar. Oder Parkkommando ausführen <ESC>sxl<CR>
WARNUNG #162 bis #164	***Wird zur Zeit nicht verwendet !***	
HINWEIS #165 Papierende	Papierende erkannt	Papier neu einlegen oder Papiertransport überprüfen. Hinweis nur mit neu einlegen von Papier (<ESC>s[]-/+<CR> rücksetzbar.
WARNUNG #166 bis #199	***Wird zur Zeit nicht verwendet !***	

Fehlerebene 4 - Hardware

Meldung	Mögliche Ursachen	Maßnahmen
Fehler #200 bis #202	***Wird zur Zeit nicht verwendet !***	
FEHLER #203 Papierstau	Papierstau erkannt	Papiertransport überprüfen (Endlospapier eingelegt und auf Etikett eingestellt) Zurücksetzen der Fehlernummer nur mit neu einlegen von Papier (<ESC>s[]-/+<CR> rücksetzbar.
Fehler #204 bis #206	***Wird zur Zeit nicht verwendet !***	
FEHLER #207	Fehler im Feeder. Fehler bei Ausfädeln.	Feeder Kontrollieren. Erst nach einem kompletten Zyklus(Schachtwechsel, oder Park -> Feed) wird der Fehler zurück gesetzt.
FEHLER #208	Fehler im Feeder. Fehler bei Ausfädeln.	Feeder Kontrollieren. Erst nach einem kompletten Zyklus(Schachtwechsel, oder Park -> Feed) wird der Fehler zurück gesetzt.
FEHLER #209	Fehler im Feeder. Papier wurde nach Entnahmekommando <ESC>s=[1..4=]->CR> nicht entfernt.	Feeder Kontrollieren. Papier aus Schacht entnehmen.
FEHLER #210	Auswertung nur wenn Bit5 in erweiterten K-Parameter gesetzt ist. Fehler bei Ausmessung der Synchronisationsmarken bei Errorrecover. Die Angaben die bei der Initsequenz (<ESC>i..) gesetzt wurden, stimmen mit dem Papier nicht überein.	Papier Prüfen. Errorrecover durchführen <ESC>!<HT>. Evt. falsche Synchronisationsmarken. Prüfen der Werte die mit der Initsequenz <ESC>i.. gesetzt wurden. Eventuell ist Papier falsch eingelegt. (Papier muß mit bestimmter Schnittposition eingelegt sein). Mit FF (<ESC>z2<CR>) kann auf richtige Schnittposition positioniert werden.
FEHLER #211	Auswertung nur wenn Bit5 in erweiterten K-Parameter gesetzt ist. Fehler Papierweg. Bei der Messung des Papierwegs zwischen Cutter und Sensor wurde falscher Wert ermittelt.	Eventuell Cutter Problem, Stepper hat blockiert oder Sensor Problem. Papier Prüfen, Errorrecover durchführen.
Fehler #212 bis #214	***Wird zur Zeit nicht verwendet !***	

Meldung	Mögliche Ursachen	Maßnahmen
FEHLER #215	Auswertung nur wenn Bit5 in erweiterten K-Parameter gesetzt ist. Fehler bei Ausgabe von Ticket. Hier wurde keine Bestromung auf dem Ticket durchgeführt.	Papier Prüfen. Errorrecover durchführen <ESC>!<HT>durchführen.
Fehler #216 bis #239	***Wird zur Zeit nicht verwendet !***	
HARDWARE #240 Abschneider	Abschneider funktioniert nicht.	Prüfen, ob Abschneider angeschlossen ist oder blockiert.

iii. USB-Support

Der Drucker kommuniziert mit dem Host PC über eine USB2.0 Schnittstelle. Innerhalb des Druckers wird ein spezieller USB-Controllerbaustein (FT8BM245) verwendet.
Um mit dem Drucker über USB Kommunizieren zu können, ist es notwendig einen entsprechenden Treiber zu Installieren.
Der Verwendetet USB-Treiber setzt voraus, dass seitens des Hosts ein virtueller COM-Anschluss verwendet wird.
Dieses bedeutet das aus sicht des Druckertreibers nicht ein USB-Port für die Druckerkommunikation sondern ein Virtueller Com-Port verwendet wird (siehe Bild: Datenfluss Druckdaten).
Gleichzeitig kann man mit dem Drucker über einen Druckertreiber kommunizieren, siehe hierzu Dokument „PDM170_Druckertreiber.pdf“.

Applikation
GDI
PDM Printer Driver DLL
Spooler
PDM Language Monitor
Bidirektionaler Thread
Pipe Server
Port Monitor
Pipe:
\\.\PIPE\PDM\xxx
wobei xxx das Device der Schnittstelle entspricht. z.B. Com6
\\.\PIPE\PDM\Com6
Device Treiber
Applikation oder Überwachungsprogramm.
Für Drucker, Steuerung, Überwachung, Konfiguration.
- ZOC
- User DLL
- ...
Printer

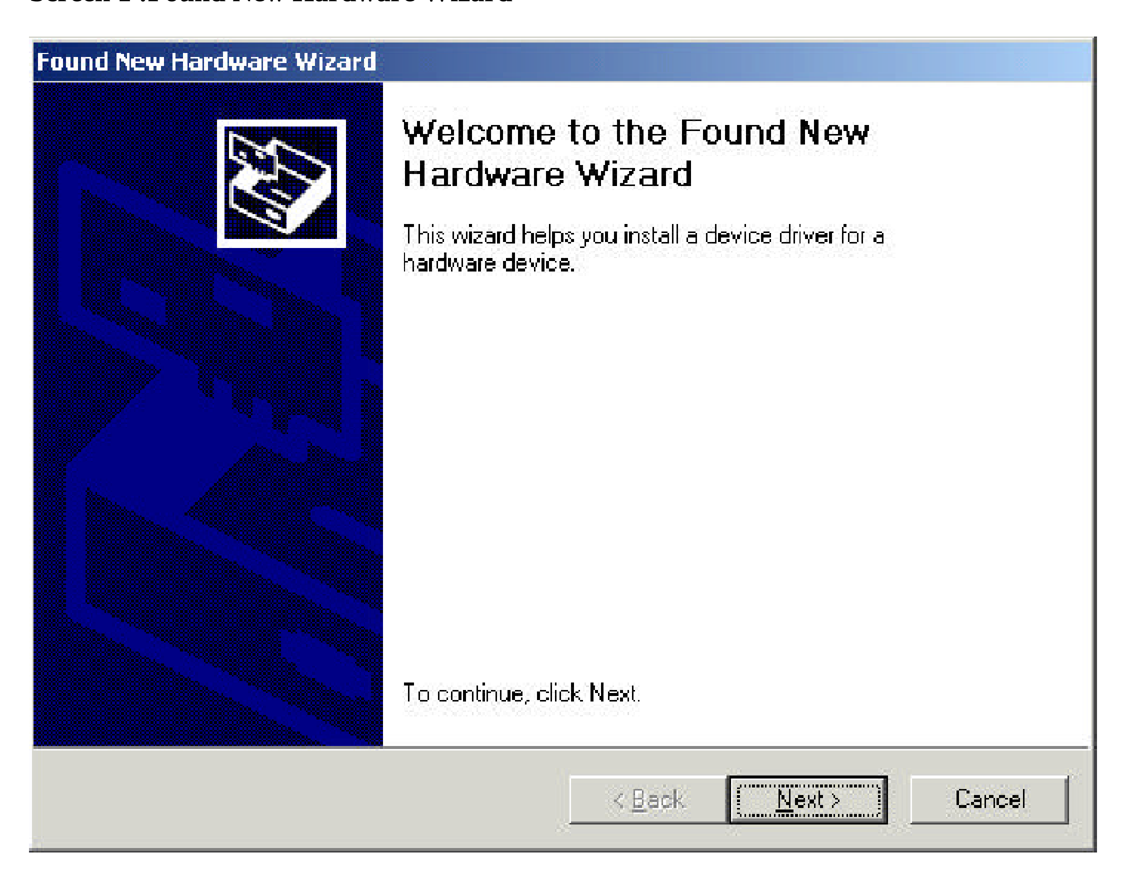
Bild: Datenfluss Druckdaten

Installationsanleitung des Treibers für FT8U232/245 Bausteine unter Windows 2000

Diese Anleitung setz vorraus, dass auf dem zu Installierenden System sich bisher kein FTDI-Treiber befindet.
So bald der Drucker über die USB-Schnittstelle mit dem Host verbunden wird, erscheint der "Hardware Wizard" auf dem Desktop.

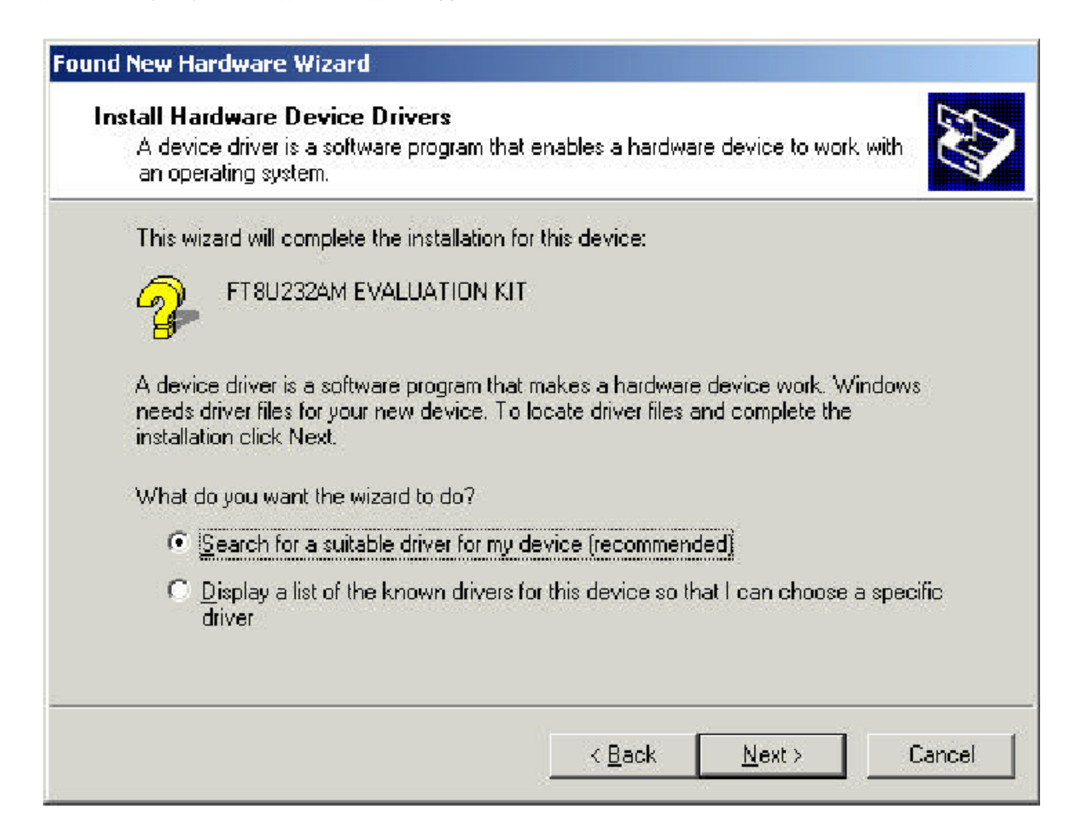
Bitte beachte Sie dass die Screenshots für die Installation des Bausteins FT8U232AM entsprechen, die Installation für den Baustein FT8U245AM ist ähnlich.
Auch entsprechen die Screenshots einer Englischeninstallation. Für eine deutschsprachige Installation werden die entsprechenden Deutschen Texte angezeigt.

Screen 1 :Found New Hardware Wizard



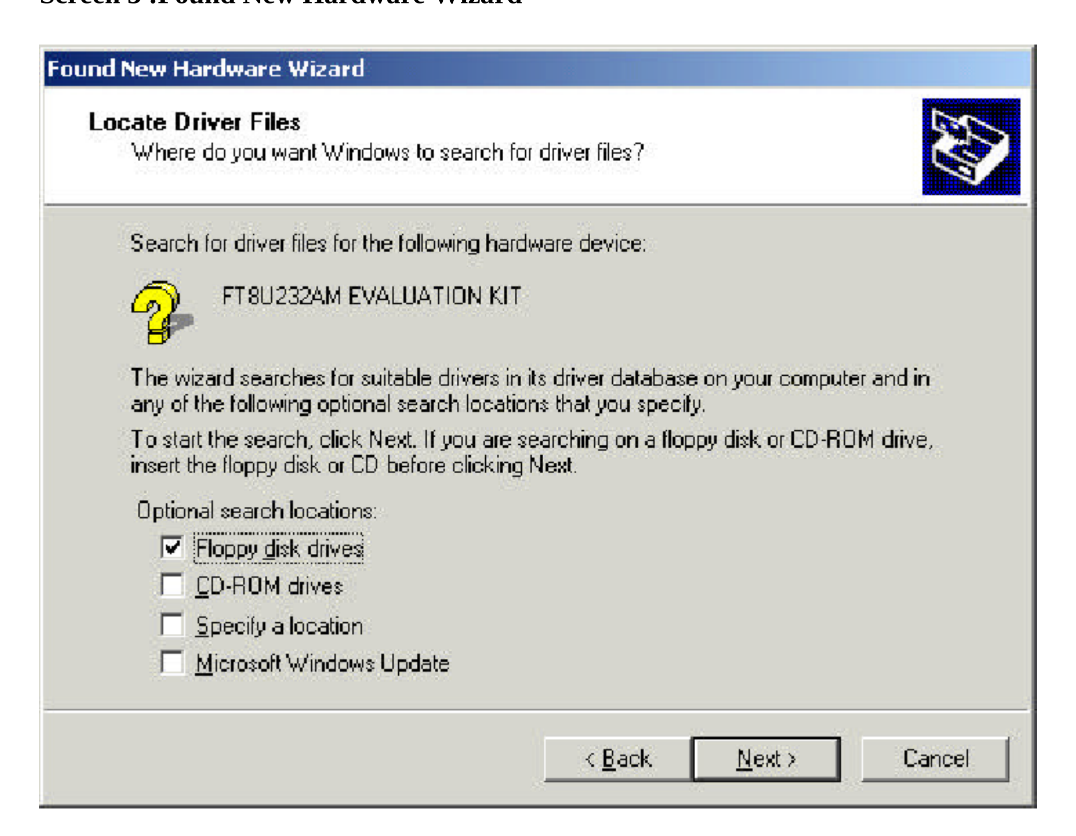
Button "Next" oder "Weiter" betäigen:

Screen 2 :Found New Hardware Wizard



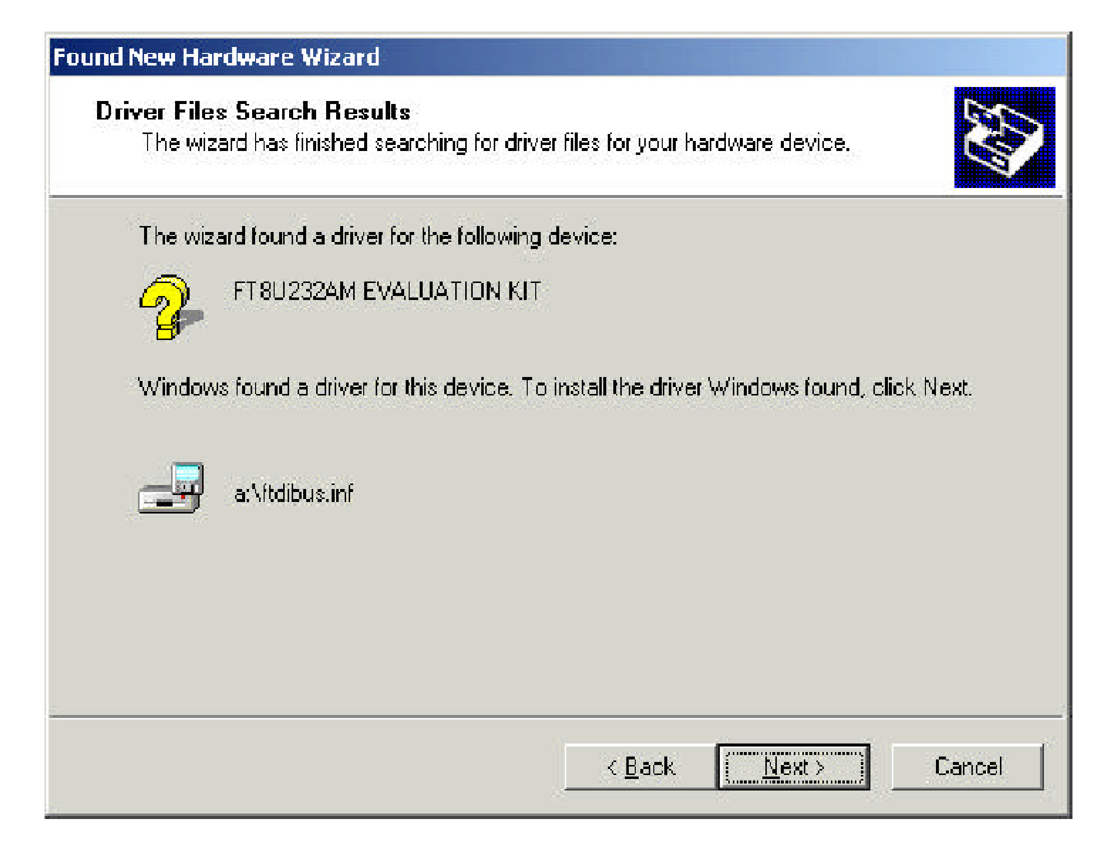
Button “Next” oder “Weiter” betäigen:

Screen 3 :Found New Hardware Wizard



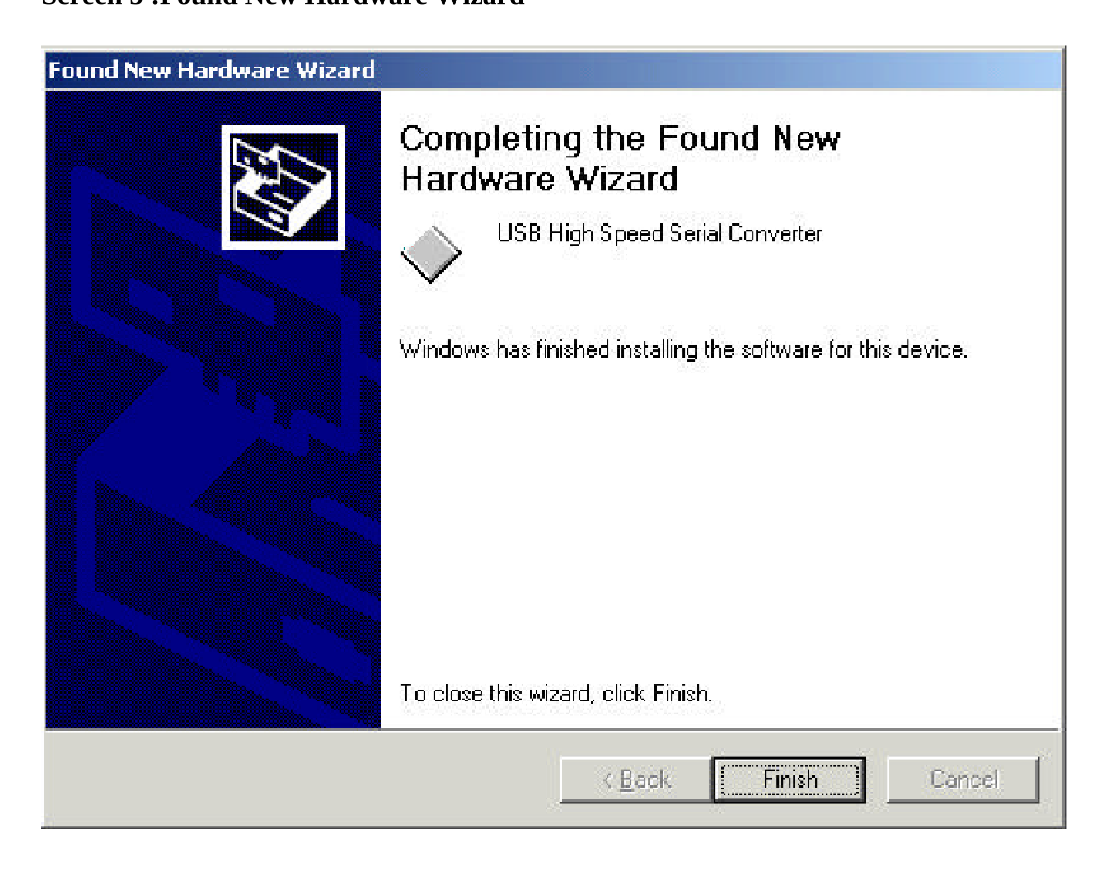
Button “Next” oder “Weiter” betäigen:

Screen 4 :Found New Hardware Wizard



Button “Next” oder “Weiter” betäigen:

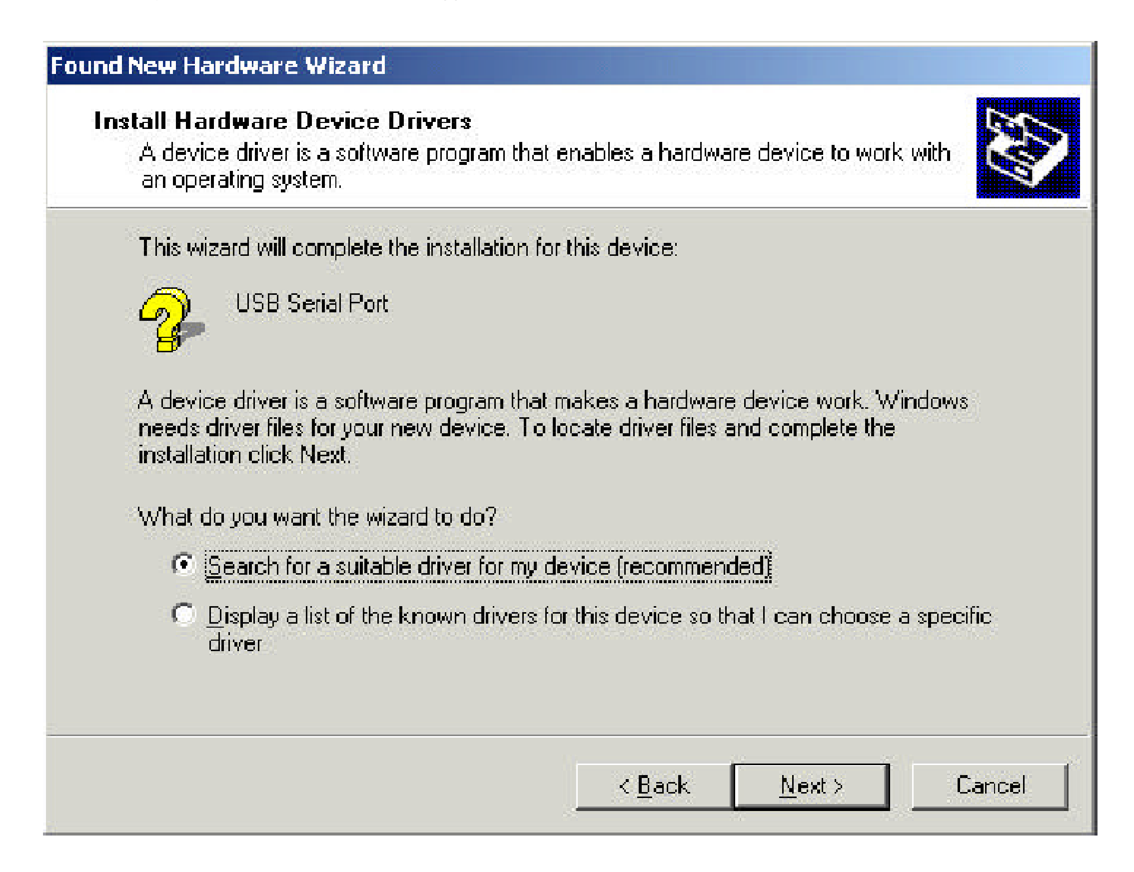
Screen 5 :Found New Hardware Wizard



Button “Finish” oder “Fertig” betäigen:

Durch Betätigen der Taste wird der erste Teil der Ínstallation des USB zu Serial Konverter installiert . Der zweite Teil der Installation gleicht der Installation einer Seriellenschnittstelle. Dieses sollte automatisch nach der Installation des ersten Teils erfolgen.

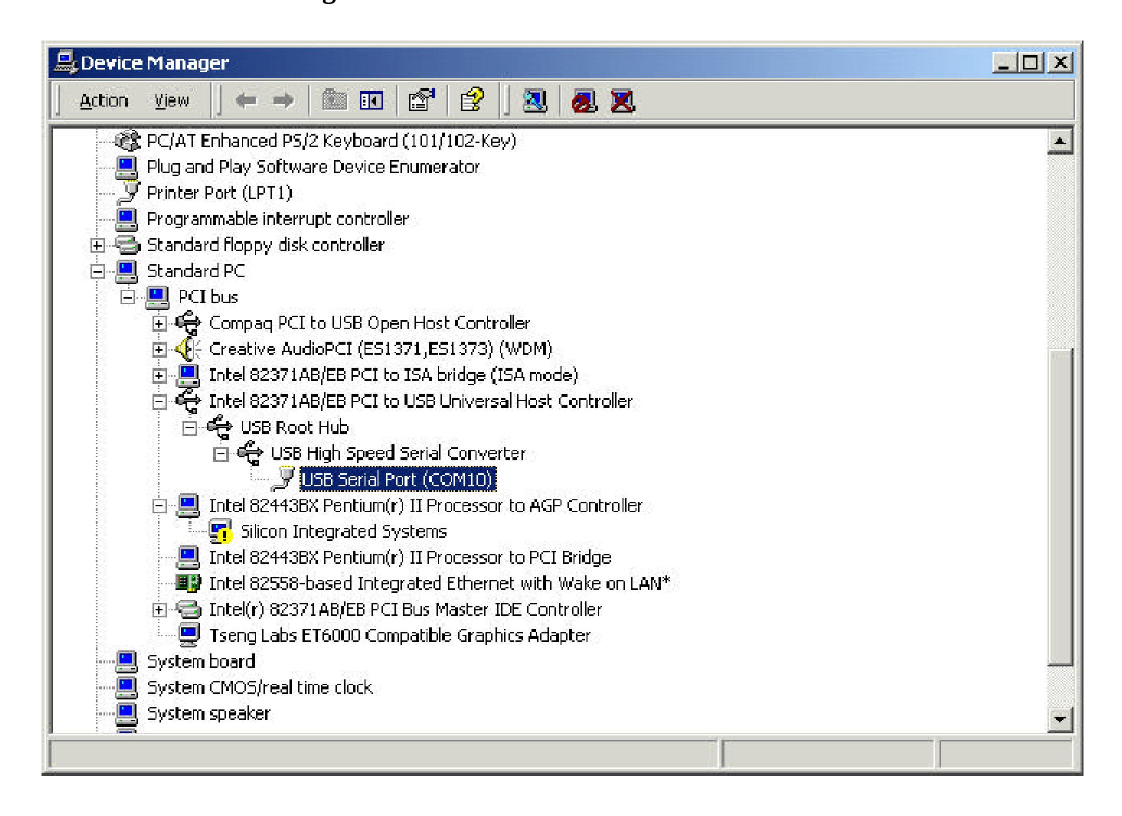
Screen 6 :Found New Hardware Wizard



Nach dem die Installation des Seriellenports abgeschlossen ist, soll man die Installation überprüfen. Dazu öffnet man den Device Manager.
Der Treiber sollte als ein „USB Serial Port“ installiert sein.
(COMx) ist nun verbunden mit dem USB High Speed Serial Converter.
(Einstellungen/Systemsteuerung/System/Geräte-Manager)
Der nochfolgende Screen zeigt die Einstellungen nach der Betätigung von Systemsteuerung -> System -> Hardware -> Geräte-Manager (Control Panel -> System Properties -> Hardware -> Device Manager),

dann Ansicht -> Geräte nach Verbindung .

Screen 7 : Device Manager



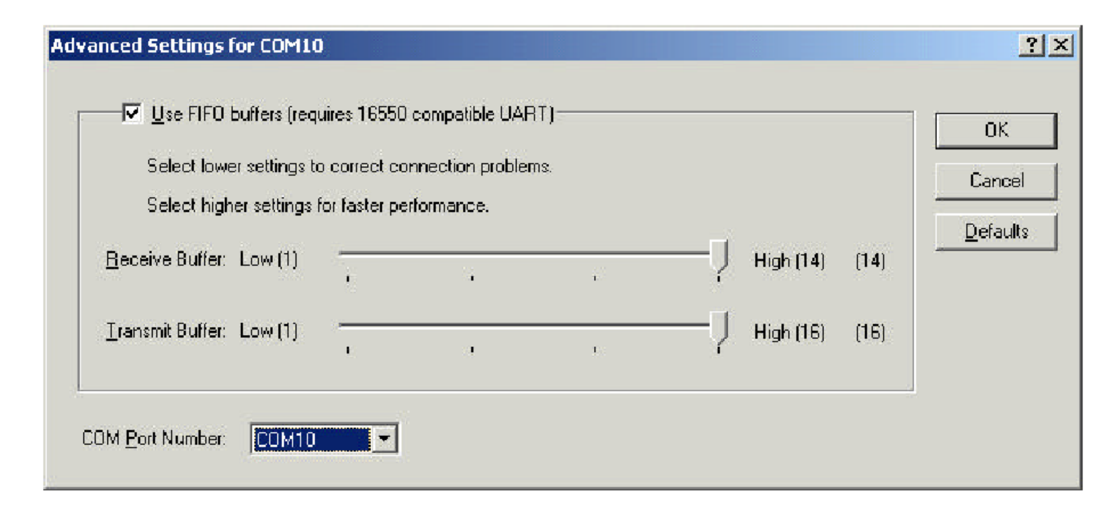
Um die virtuelle COM Port Eigenschaften zu Ändern, selektieren Sie den USB Serial Port und dann auf Eigenschaften klicken.
Dieses erlaubt Ihnen die Einstellungen des seriellen Ports zu ändern, wie Baudrate, Daten Bits, usw.
Genauso kann hier die Nummer des COM-Ports geändert werden.

Screen 8 : USB Serial Port (COM19) Properties

USB Serial Port (COM10) Properties
General | Port Settings | Driver
USB Serial Port (COM10)
Device type: Ports (COM & LPT)
Manufacturer: FTDI
Location: on USB High Speed Serial Converter
Device status
This device is working properly.
If you are having problems with this device, click Troubleshooter to start the troubleshooter.
Troubleshooter...
Device usage:
Use this device (enable)
OK | Cancel

Wählen Sie nun „Port Settings“ aus.
Um die Schnittstellennummer Ändern zu können betätigen sie das Dropdown-Feld und wählen nun den COM-Port Number.
Siehe Secreen 9.

Screen 9 : Advanced Settings for COM10



Wenn man auf den Geräte Manager Screen zurück kehrt kann man nun die neuen Einstellungen der COM Portnummer sehen.

Screen 10 : Device Manager

iv : Spezifikation

Steckerbelegung:

Schnittstellenanschlüsse Druckerseite RJ45

Pin	Signal	Sender	Funktion
1,2	5V		5 Volt Versorgungsspannung
3	CTS	Computer	Computer bereit für Datenempfang
4	RTS	Printer	Übertragungsanforderung, immer HIGH
5	RxD	Computer	Daten empfangen. Empfängt die vom Computer übertragenen Daten.
6	TxD	Printer	Daten senden. Sendet die Daten an den Computer.
7,8	GND		Signalerde

Versorgungsstecker

Pin	Signal	Funktion
1,4	24V	24 Volt Versorgungsspannung
2,3	GND	Masse Anschluß

Technische Daten

Siehe Dokumentation „**datenblatt_PDM170.pdf**"